PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Efficient sampling-based Bayesian Active Learning for synaptic characterization

Camille Gontier[1,2]*, **Simone Carlo Surace**[1], **Igor Delvendahl**[3,4], **Martin Müller**[3,4,5], **Jean-Pascal Pfister**[1]

1 Department of Physiology, University of Bern, Bern, Switzerland, **2** Rehab Neural Engineering Labs, University of Pittsburgh, Pittsburgh, Pennsylvania, United States of America, **3** Department of Molecular Life Sciences, University of Zurich, Zurich, Switzerland, **4** Neuroscience Center Zurich, University of Zurich, Zurich, Switzerland, **5** University Research Priority Program (URPP), Adaptive Brain Circuits in Development and Learning (AdaBD), University of Zurich, Zurich, Switzerland

* camille.gontier@unibe.ch

Abstract

Bayesian Active Learning (BAL) is an efficient framework for learning the parameters of a model, in which input stimuli are selected to maximize the mutual information between the observations and the unknown parameters. However, the applicability of BAL to experiments is limited as it requires performing high-dimensional integrations and optimizations in real time. Current methods are either too time consuming, or only applicable to specific models. Here, we propose an Efficient Sampling-Based Bayesian Active Learning (ESB-BAL) framework, which is efficient enough to be used in real-time biological experiments. We apply our method to the problem of estimating the parameters of a chemical synapse from the postsynaptic responses to evoked presynaptic action potentials. Using synthetic data and synaptic whole-cell patch-clamp recordings, we show that our method can improve the precision of model-based inferences, thereby paving the way towards more systematic and efficient experimental designs in physiology.

Author summary

Optimizing the design of an experiment is a critical problem in biology. However, most experiments still rely on suboptimal designs, which may not yield sufficient information about the studied system. Consequently, such experiments often require more observations to reach a certain result. An efficient theoretical framework to alleviate this issue is called Optimal Experiment Design (OED), in which experimental protocols are selected to reduce the uncertainty of inferred parameters. However, the applicability of OED methods to actual experiments is limited: they often require computations which are too long for sequential experiments, and do not generalize to different models. Here, we developed a method called Efficient Sampling-Based Bayesian Active Learning (ESB-BAL), and apply it to the problem of estimating the parameters of a chemical synapse from its evoked postsynaptic currents. After each new observation, the optimal next stimulation time can be computed using ESB-BAL. Using recordings in cerebellar brain slices, we show that our method is fast enough to be used in real-time biological experiments

OPEN ACCESS

Citation: Gontier C, Surace SC, Delvendahl I, Müller M, Pfister J-P (2023) Efficient sampling-based Bayesian Active Learning for synaptic characterization. PLoS Comput Biol 19(8): e1011342. https://doi.org/10.1371/journal.pcbi.1011342

Editor: Ulrik R. Beierholm, Durham University, UNITED KINGDOM

Received: January 4, 2023

Accepted: July 10, 2023

Published: August 21, 2023

Data Availability Statement: JULIA files are available in the following package: https://github.com/Theoretical-Neuroscience-Group/BinomialSynapses.jl Data and analysis files are available in the following package: https://github.com/camillegontier/data_analysis_ESBBAL.

Funding: The work presented in this paper was supported by the Swiss National Science Foundation (https://www.snf.ch/en) under grant number 31003A_175644 entitled "Bayesian Synapse" and received by JPP, and under grant number P500PM_210800 entitled "Improving the precision, stability and robustness of Brain-Computer Interfaces with Bayesian inference" and received by CG. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

and can significantly reduce the uncertainty of inferred parameters. Our method can be readily used by experimentalists via a simple interface. Moreover, our proposed solution is general enough to be applicable to different experimental settings.

Introduction

In neuroscience, machine learning, and statistics, a central problem is that of inferring the parameters θ of a model $\mathcal{M}$. For instance, in supervised learning, one may want to learn the parameters of a Deep Neural Network (DNN) so as to minimize the difference between its output and training labels; in this case, $\mathcal{M}$ represents the DNN to be trained, and θ represents its weights and biases. Similarly, in biology, the parameters of a system can be studied by fitting a biophysical model to recorded observations. In most cases, these parameters can be neither directly measured nor analytically computed, but can be inferred using the recorded outputs of the system y as a response to input stimuli x. In biology, the physical quantities of a system (e.g. an organ, a cell, or a synapse) can be estimated by deriving a generative biophysical model $\mathcal{M}$ of the system, and by fitting its parameters θ to the observed responses y to experimental inputs x. By computing the likelihood of the outputs given the inputs and the parameters $p(y|x, \theta)$, it is possible to obtain either a point-based estimate of the parameters such as the maximum likelihood parameters θ_{ML} or the maximum a posteriori parameters θ_{MAP} [1], or to compute the full posterior distribution $p(\theta|x, y) \propto p(y|x, \theta)$ using for instance the Metropolis-Hastings (MH) algorithm [2].

However, the accuracy of these estimates critically depends on the pair (x, y), and especially on how the successive input stimuli $x = x_{1:T}$ are chosen. For instance, training a DNN on non independent and identically distributed (i.i.d.) training examples (i.e. blocked training) will lead to catastrophic forgetting [3]. On the other hand, most experiments in biology still rely on pre-defined and non-adaptive inputs $x_{1:T}$, which may not yield sufficient information about the true parameters of the studied system. Consequently, experiments often require more observations or repetitions to reach a certain result, which increases their cost, time, and need for subjects.

An efficient framework to alleviate this issue is called Bayesian Active Learning (BAL). Knowing the current estimate of the parameters, the experimental protocol (i.e. the next input x_{t+1}) can be optimized on the fly to maximize the mutual information between the recordings and the parameters (Fig 1C). BAL is a branch of Optimal Experiment Design (OED) theory [4–6]. It has already been used in neuroscience to infer the parameters of a Generalized Linear Model (GLM) [7], the nonlinearity in a linear-nonlinear-Poisson (LNP) encoding model [8], the receptive field of a neuron [9], or the parameters of a Hidden Markov Model (HMM) [10].

However, implementing BAL for biological settings can be challenging, especially for real-time applications. Its applicability to real experiments is limited by two main drawbacks. Firstly, it requires computing an update of the posterior distribution of parameters after each time step, and using it to compute the expected information gain from future experiments. This involves solving an optimization problem over a possibly high-dimensional stimulus space: current methods are either too time consuming, or only applicable to specific models. Secondly, to reduce computational complexity, classical implementations of BAL usually only optimize for the immediate next stimulus input. This classical myopic approach disregards all future observations in the experiment, and is thus possibly sub-optimal [6, 11, 12].

Our main contribution is to provide a general framework for approximate online active learning, called Efficient Sampling-Based Bayesian Active Learning (ESB-BAL). We use

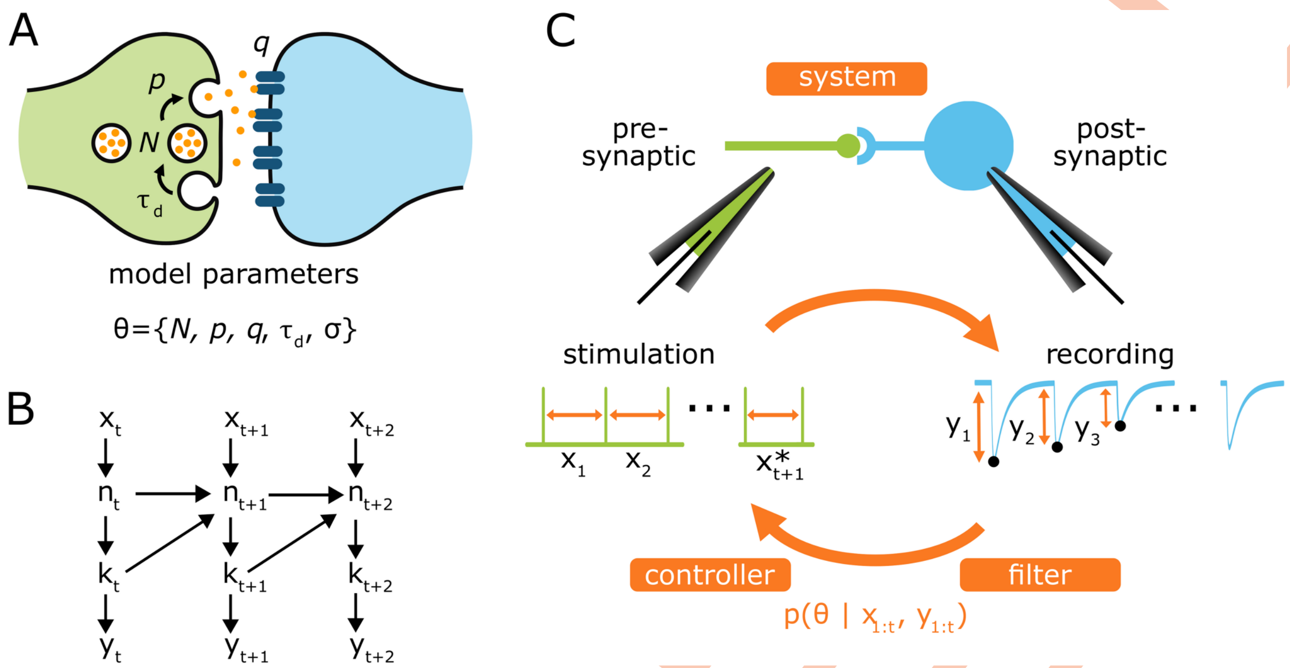

Fig 1. A: Model of binomial synapse with STD. In chemical synapses, the presynaptic terminal is characterized by the presence of N vesicles containing the neurotransmitter molecules, n_t of them being in the readily-releasable state [23]. Upon the arrival of a presynaptic spike, these vesicles will stochastically fuse with the plasma membrane and release their neurotransmitters into the synaptic cleft. After spike t, k_t vesicles (out of the n_t available ones in the readily-releasable pool) release their neurotransmitters with a probability p. Neurotransmitters will bind to postsynaptic receptors: a single release event triggers a quantal response q. The total recorded postsynaptic current y_t (i.e. the output of the system) is the sum of the effects of the k_t release events. After releasing, vesicles are replenished with a certain time constant τ_D, which determines short-term depression. B: Modelisation of the synapse an an IO-HMM [10]. C: Bayesian Active Learning applied to biology. At each time step, the response of the **system** (e.g. here a synapse) to artificial stimulation is recorded. This observation y_t is used by the **filter** to compute the posterior distribution of parameters $p(\theta|x_{1:t}, y_{1:t})$. Given this posterior, the **controller** then computes the next input x^*_{t+1} to maximize the expected gain of information of the next observation. In classical experiment design, the inputs $x_{1:T}$ are defined and fixed prior to the recordings.

https://doi.org/10.1371/journal.pcbi.1011342.g001

particle filtering, which is a highly versatile filtering method [13], for posterior computation; and propose a parallel computing implementation [14, 15] for efficient posterior update and information computation. Whereas previous implementations of active learning either relied on time consuming Monte Carlo (MC) methods [16, 17] or were only applicable to special cases, such as linear models or GLM [7], our proposed solution is fast enough to be used in real-time biological experiments and can be applied to any state-space model.

To illustrate our method, we apply it to the problem of inferring the parameters of a chemical synapse with Short-Term Depression (STD). Upon the arrival of a presynaptic action potential, vesicles from a pool of N independent release sites will fuse with the presynaptic plasma membrane with a probability p, each of these release events giving rise to a quantal current q [18, 19]. In addition, synaptic transmission is also dynamic. Short-term depression occurs when the inter-stimulation interval (ISI) is shorter than the time needed for synaptic vesicle replenishment [20]. A synapse exhibiting STD can thus be described by its parameters N, p, q, and by its depression time constant. These parameters can be inferred using excitatory postsynaptic currents (EPSCs) recorded from the postsynaptic cell and elicited by stimulating the presynaptic axon. The accuracy of these estimates critically depends on the presynaptic stimulation times: if inter-stimulation intervals are longer than the depression time constant, STD will not be precisely quantified. But if the stimulation frequency is too high, the pool of presynaptic vesicles will be depleted, leading to poor parameter estimates [21, 22]. Synaptic characterization is thus a relevant example application for ESB-BAL, as it requires careful tuning of the inputs $x_{1:T}$, which in this case correspond to the inter-stimulation intervals (i.e. input x_t is the time interval between stimulation numbers $t - 1$ and t); but it is also a challenging one: computation needs to be faster than the typical ISI, which can be on the order of a few milliseconds. Using synthetic data, we show that our method allows to significantly reduce the uncertainty of the estimate in comparison to classically used non-adaptive stimulation protocols. We also show that the rate of information gain (in bit/s) of the whole experiment can be optimized by adding a penalty term for longer ISIs. Lastly, we extend active learning to non-myopic designs. Using recordings from cerebellar mossy fiber to granule cell synapses from acute mouse brain slices, we show that our framework is sufficiently efficient for optimizing not only the immediate next stimulus, but rather the future stimuli in the experiment.

Results

A general setting for Bayesian Active Learning

When using active learning in sequential experiments, three key elements need to be defined (Fig 1C):

1. The **system** to be studied: it is described by a generative model $\mathcal{M}$, which parameters θ can be inferred from its observed responses $y_{1:T}$ to a set of T input stimuli $x_{1:T}$. Given the stochastic nature of most systems studied in biology, the random variable $Y_{1:T}$ corresponding to the observations can take various values $y_{1:T}$ according to a distribution $p(y_{1:T}|x_{1:T}, \theta)$. In our application example of BAL, the system will be a model of binomial neurotransmitter release (see Section The system: A binomial model of neurotransmitter release).
2. A **filter** that computes the posterior distribution of the parameters given the previous inputs and observations $p(\theta|x_{1:t}, y_{1:t})$: after each new input x_{t+1} and observation y_{t+1}, it is updated to obtain $p(\theta|x_{1:t+1}, y_{1:t+1})$ (see Section The filter: Online computation of the posterior distributions of parameters).
3. A **controller** that computes the next optimal input stimuli x^*_{t+1} so as to maximize a certain utility function, which is often defined as the mutual information between the parameter random variable Θ and the response random variable Y_{t+1} given the experimental inputs $I_{x_{t+1}}(\Theta; Y_{t+1}|h_t)$, where $h_t = (x_{1:t}, y_{1:t})$ is the experiment history (see Section The controller: Computation of the optimal next stimulation time).

Throughout the paper, we use upper-case notations for random variables, and lower-case notations for the specific values they might take (Table 1). For instance, Θ is the random variables corresponding to the hidden parameters of the system, while θ describes a specific value of these parameters. Similarly, Y_t is the random variable corresponding to the output of the system at time step t, while y_t is the actual value observed at time step t. Hence, $H(\Theta)$ will represent the entropy of the random variable Θ, while $p(\Theta = \theta)$ is the probability of Θ taking the specific value θ. For simplicity, we will often use the shorter notation $p(\theta)$ for $p(\Theta = \theta)$. This distinction is crucial e.g. in Eq 3, to distinguish the conditional entropy $H_{x_{t+1}}(\Theta|h_t, Y_{t+1}) = \int dy_{t+1} p(y_{t+1}|h_t, x_{t+1}) H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = y_{t+1})$ from its value given a specific realization of the future observation $H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = y_{t+1})$.

In synaptic characterization, inputs correspond to a set of T stimulation times $x_{1:T}$ and observations correspond to recorded excitatory postsynaptic currents (EPSCs) $y_{1:T}$. In case of successive experiments [9], the mutual information between the parameters and the next observation Y_{t+1} conditioned on the experiment history h_t is:

$$I_{x_{t+1}}(\Theta; Y_{t+1}|h_t) = H(\Theta|h_t) - H_{x_{t+1}}(\Theta|h_t, Y_{t+1}) \quad (1)$$

Table 1. Notations.

Indices:	
$t \leq T$	Number of observations
$i \leq M_{\text{out}}$	Number of outer particles
$j \leq M_{\text{in}}$	Number of inner particles
Parameters:	
N	Number of presynaptic independent release sites [-]
p	Release probability upon the arrival of a presynaptic spike [-]
q	Quantum of postsynaptic current elicited by one release event [A]
σ	Standard deviation of the recording noise [A]
τ_D	Time constant of synaptic vesicle replenishment [s]
Random variables:	
θ	Vector of unknown parameters
Y_t	Output of the system at time t
Functions:	
$p_\theta(\cdot)$	Probability distribution conditioned on $\Theta = \theta$
$I(\cdot;\cdot)$	Mutual information
$H(\cdot)$	Differential entropy
Others:	
x_t	Input to the system at time step t (t^{th} inter-spike interval)
y_t	Recording at time step t (t^{th} EPSC amplitude)
h_t	History of observations ($x_{1:t}$, $y_{1:t}$)
$\mathcal{M}$	Generative model of the studied system
n_t	Number of vesicles in the readily-releasable state immediately
before spike t	
k_t	Number of released vesicles after spike t

https://doi.org/10.1371/journal.pcbi.1011342.t001

where $H(\Theta|h_t)$ is the entropy of Θ given the experiment history up to time step t:

$$H(\Theta|h_t) = -\int d\theta p(\theta|h_t) \log p(\theta|h_t) \tag{2}$$

and

$$H_{x_{t+1}}(\Theta|h_t, Y_{t+1}) = \int dy_{t+1} p(y_{t+1}|h_t, x_{t+1}) H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = y_{t+1}) \tag{3}$$

is the conditional entropy of Θ given the future observation random variable Y_{t+1}. Since the actual value of the future observation is unknown, we take the average over y_{t+1} of the conditional entropy $H_{x_{t+1}}(\Theta|h_{,} Y_{t+1} = y_{t+1})$ conditioned on a certain value y_{t+1}. As the predictive distribution depends on the unknown parameters, we also have to take an average over θ, using the current posterior distribution $p(\theta|h_t)$ at time t [7]:

$$p(y_{t+1}|h_t, x_{t+1}) = \int d\theta p(y_{t+1}|h_t, x_{t+1}, \theta) p(\theta|h_t) \tag{4}$$

The goal of Bayesian active learning is to select the next stimulation to maximize the mutual information between the parameters and all future observations:

$$x^*_{t+1} = \arg \max_{x_{t+1} \in \mathcal{S}_{t+1}} \max_{n} \max_{x_{t+2:t+n} \in \mathcal{S}_{t+2:t+n}} I_{x_{t+1:t+n}}(\Theta; Y_{t+1:t+n}|h_t) \tag{5}$$

where $\mathcal{S}_{t+1}$ is the set of possible inputs at time step $t + 1$ and $\mathcal{S}_{t+2:t+n}$ is the set of possible protocols for the stimulations from time step $t + 2$ to $t + n$. This set of protocols includes all the stimulation constraints, e.g. the remaining time of the experiment or the minimal inter-stimulation time. Optimizing all future inputs is an intractable problem (especially for online applications), since the algorithmic complexity scales exponentially with the number of observations n. For this reason, BAL only optimizes for the next stimulus (an approach referred to as a *myopic* design) (see Fig 1C):

$$x^*_{t+1} = \underset{x_{t+1}\in\mathcal{S}_{t+1}}{\arg\max}\; I_{x_{t+1}}(\Theta; Y_{t+1}|h_t) \tag{6}$$

Different methods have been proposed to compute Eq 6. Monte Carlo (MC) methods [16] or a variational approach [17] can be employed, but they usually require long computation times that can be impractical if the time between successive experiments is short. Closed-form solutions or approximations can be computed only for some special cases, such as linear models or GLM [7].

The system: A binomial model of neurotransmitter release

To illustrate our ESB-BAL framework, we apply it to the problem of estimating the parameters of a chemical synapse, represented as a state-space model with unobservable hidden states and input-dependent state transitions. A classical used model to describe the release of neurotransmitters at chemical synapses is called the binomial model [1, 2, 19–21, 24, 25]. According to this model, a synapse is described as an Input-Output Hidden Markov Model (IO-HMM [10]) with the following parameters (units are given in square brackets, see also Fig 1A):

- N (the number of presynaptic independent release sites [-]);
- p (their release probability upon the arrival of a presynaptic spike [-]);
- q (the quantum of current elicited in the postsynaptic cell by one release event [A]);
- σ (the standard deviation of the recording noise [A]);
- τ_D (the time constant of synaptic vesicle replenishment [s]).

The variables n_t and k_t represent, respectively, the number of available vesicles in the readily-releasable state at the moment of spike t (with $0 \leq n_t \leq N$), and the number of vesicles (among n_t) released after spike t (with $0 \leq k_t \leq n_t$). For simplicity, we use the notations $p_\theta(\cdot) = p(\cdot|\theta)$ with $\theta = [N, p, q, \sigma, \tau_D]$, and $z_t \coloneqq (n_t, k_t)$ to refer to the hidden variables at time step t.

To summarize, the synapse is modelled as an IO-HMM, where:

- The input x_t refers to the time interval since the previous stimulation, i.e., to the inter-spike interval;
- The hidden state variable $z_t \coloneqq (n_t, k_t)$ encompasses both the number of available vesicles immediately before spike t ($n_t \in \{0, \ldots, N\}$) and the corresponding number of released vesicles ($k_t \in \{0, \ldots, n_t\}$);
- The observable variable y_t is the recorded value of the postsynaptic current due to spike t. De facto, the postsynaptic is continuously monitored: the specific value y_t is computed as the peak amplitude following spike t (see Materials and methods).

The probability of recording a set of T EPSCs $p_\theta(y_{1:T})$ is computed as the marginal of the joint distribution of the observations $y_{1:T}$ and the hidden variables $z_{1:T}$, i.e. $p_\theta(y_{1:T}) = \sum_{z_{1:T}} p_\theta(y_{1:T}, z_{1:T})$, where the joint distribution $p_\theta(y_{1:T}, z_{1:T}) = p_\theta(y_{1:T}, n_{1:T}, k_{1:T})$ is given by

$$p_\theta(y_{1:T}, n_{1:T}, k_{1:T}) = p_\theta(y_1|k_1)p_\theta(k_1|n_1)p_\theta(n_1)\prod_{t=2}^{T}p_\theta(y_t|k_t)p_\theta(k_t|n_t)p_\theta(n_t|n_{t-1}, k_{t-1}, x_t) \tag{7}$$

where

$$p_\theta(y_t|k_t) = \mathcal{N}(y_t; qk_t, \sigma^2) \tag{8}$$

is the emission probability, i.e. the probability to record output y_t knowing that k_t vesicles released neurotransmitter; $p_\theta(k_t|n_t)$ is the binomial distribution and represents the probability that, given n_t available vesicles, k_t of them will indeed release neurotransmitter:

$$p_\theta(k_t|n_t) = \binom{n_t}{k_t}p^{k_t}(1-p)^{n_t-k_t} \tag{9}$$

Finally, $p_\theta(n_t|n_{t-1}, k_{t-1}, x_t)$ represents the process of vesicle replenishment. During the time interval x_t, each empty vesicle can refill with a probability $\pi(x_t) = 1 - \exp\left(-\frac{x_t}{\tau_D}\right)$ such that the transition probability $p_\theta(n_t|n_{t-1}, k_{t-1}, x_t)$ is given by:

$$p_\theta(n_t|n_{t-1}, k_{t-1}, x_t) = \binom{N - n_{t-1} + k_{t-1}}{n_t - n_{t-1} + k_{t-1}}\pi(x_t)^{n_t - n_{t-1} + k_{t-1}}(1 - \pi(x_t))^{N - n_t} \tag{10}$$

One can note that $n_t = n_{t-1} - k_{t-1} + v_t$, where $v_t \sim \text{Bin}(N - n_{t-1} + k_{t-1}, \pi(x_t))$ is the number of refilled vesicles during the time interval x_t. Eqs 7 to 10 define the observation model of the studied system (see Fig 1), i.e. the probability of a set of observations $y_{1:T}$ given a vector of stimuli $x_{1:T}$ and a vector of parameters θ.

The filter: Online computation of the posterior distributions of parameters

To be applicable for online experiments, the filtering block, which will compute the posterior distribution of parameters $p(\theta|h_t)$, needs to satisfy two requirements:

1. It must be sufficiently versatile to be applied to different systems and models;
2. It must be online (i.e. its algorithmic complexity should not increase with the number of observations) [26].

A promising solution is particle filtering [27], and especially the Nested Particle Filter (NPF) [13]. This algorithm is asymptotically exact and purely recursive, thus allowing to directly estimate the parameters of a HMM as recordings are acquired.

The NPF relies on two nested layers of particles to approximate the posterior distributions of both the static parameters θ of the model and of its hidden states z_t. A first outer filter with M_{out} particles is used to compute the posterior distribution of parameters $p(\theta|h_t)$, and for each of these particles, an inner filter with M_{in} particles is used to estimate the corresponding hidden states z_t (so that the total number of particles in the system is $M_{\text{out}} \times M_{\text{in}}$). After each new observation, these particles are resampled based on their respective likelihoods, hence updating their posterior distributions (S1 Fig).

The NPF was originally proposed for static HMMs, in which the state transition probability $p(z_{t+1}|z_t, \theta)$ is supposed to be constant. Here, we extend it to the more general class of Input-Output Hidden Markov Models (IO-HMMs, also called GLM-HMMs in neuroscience, see [10]), in which the state transition probability at time step t depends on an external input x_t.

For instance, state transition in our model of synapse is not stationary, but depends on the ISI x_t.

The filter (Algorithm 1) relies on the following approximation to recursively compute the likelihood of each particle. Once the observation y_t has been recorded, the likelihood of particle θ_t^i, with $i \in \{1, \ldots, M_{\text{out}}\}$, depends on

$$p(\theta_t^i | y_{1:t}) \propto p(y_t | y_{1:t-1}, \theta_t^i, x_t) p(\theta_t^i | y_{1:t-1}) \quad (11)$$

with

$$p(y_t | y_{1:t-1}, \theta_t^i, x_t) = \sum_{z_{t-1:t}} p(y_t | z_t, \theta_t^i) p(z_t | z_{t-1}, \theta_t^i, x_t) p(z_{t-1} | y_{1:t-1}, \theta_t^i) \quad (12)$$

If the variance of the jittering kernel κ (which mutates the samples to avoid particles degeneracy and local solutions, see Materials and methods) is sufficiently small, and hence if $\theta_t^i \approx \theta_{t-1}^i$, the approximation $p(z_{t-1} | y_{1:t-1}, \theta_t^i) \approx p(z_{t-1} | y_{1:t-1}, \theta_{t-1}^i)$ allows to approximate Eq 12 as $p(y_t | y_{1:t-1}, \theta_t^i, x_t) \approx \sum_{z_{t-1:t}} p(y_t | z_t, \theta_t^i) p(z_t | z_{t-1}, \theta_t^i, x_t) p(z_{t-1} | y_{1:t-1}, \theta_{t-1}^i)$, and hence to recursively compute Eq 11. In practice, the different terms in Eq 12 are computed as such: $p(y_t | z_t, \theta_t^i)$ corresponds to the *Likelihood* step of Algorithm 1; $p(z_t | z_{t-1}, \theta_t^i, x_t)$ corresponds to the *Propagation* step; and $p(z_{t-1} | y_{1:t-1}, \theta_t^i)$ corresponds to the distribution of hidden states at time $t - 1$.

Contrary to previous methods for fast posterior computation that were only applicable to specific models [7], our filter can be applied to any state-space dynamical system, including non-stationary and input-dependent ones. Moreover, it does not require to approximate the posterior as a Gaussian nor require a time consuming (and possibly unstable) numerical optimization step, while being highly parallelizable and efficient [14, 15].

Algorithm 1: Particle filtering for computing one step update of the posterior distribution of parameters

```
Input: $\{\theta_{t-1}^i\}_{1 \le i \le M_{\text{out}}}$, $\{n_{t-1}^{i,j}, k_{t-1}^{i,j}\}_{1 \le j \le M_{\text{in}}}$, $x_t$, $y_t$;
for $i$ in 1 ... $M_{\text{out}}$ do
  Jittering: update the outer particles $\theta_t^i = \kappa(\theta_{t-1}^i)$;
  for $j$ in 1 ... $M_{\text{in}}$ do
    Propagation: Draw $n_t^{i,j} \sim p(n_t^{i,j} | n_{t-1}^{i,j}, k_{t-1}^{i,j}, \theta_t^i, x_t)$ and $k_t^{i,j} \sim p(k_t^{i,j} | n_t^{i,j}, \theta_t^i)$
    Likelihood: compute $\tilde{w}_t^{i,j} = p(y_t | n_t^{i,j}, k_t^{i,j}, \theta_t^i)$;
  end
  Normalization: $\tilde{w}_t^{i,j} \leftarrow \tilde{w}_t^{i,j} / \sum_j \tilde{w}_t^{i,j}$;
  Inner particles resampling: resample $\{n_t^{i,j}, k_t^{i,j}\}_{1 \le j \le M_{\text{in}}}$ based on
$\{\tilde{w}_t^{i,j}\}_{1 \le j \le M_{\text{in}}}$;
end
Compute $w_t^i = \frac{1}{M_{\text{in}}} \sum_j \tilde{w}_t^{i,j}$;
Normalization: $w_t^i \leftarrow w_t^i / \sum_i w_t^i$
Outer particles resampling: resample $\{\theta_t^i\}_{1 \le i \le M_{\text{out}}}$ and $\{n_t^{i,j}, k_t^{i,j}\}_{1 \le j \le M_{\text{in}}}$ based
on $\{w_t^i\}_{1 \le i \le M_{\text{out}}}$;
Output: $\{\theta_t^i\}_{1 \le i \le M_{\text{out}}}$, $\{n_t^{i,j}, k_t^{i,j}\}_{1 \le j \le M_{\text{in}}}$
```

The controller: Computation of the optimal next stimulation time

The objective of experiment design optimization is to minimize the uncertainty of the estimates (classically quantified using the entropy) while reducing the cost of experimentation (defined as the number of required trials, samples, or observations). The optimal next stimulus x_{t+1}^* that will maximize the mutual information (i.e. minimize the uncertainty about θ as measured by the entropy) can be written from Eqs 1, 3, and 6 as

$$x^*_{t+1} = \underset{x_{t+1}\in\mathcal{S}_{t+1}}{\arg\min} \int d\theta p(\theta|h_t) \int dy_{t+1} p(y_{t+1}|h_t, x_{t+1}, \theta) H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = y_{t+1}) \tag{13}$$

Eq 13 requires to compute two (possibly high-dimensional) integrals over θ and y_{t+1}, for which closed-form expressions only exist for specific models. To avoid long MC simulations, we propose to use mean-field computations and to replace integrals by point-based approximations. Firstly, instead of computing the full expectation over $p(\theta|h_t)$, we set θ to its MAP value $\hat{\theta}_t$. Other estimators, such as the mean posterior value $\hat{\theta}_t = \int d\theta p(\theta|h_t)\theta$ (which can be conveniently approximated as $\hat{\theta}_t \approx \frac{1}{M_{\text{out}}}\sum_{i=1}^{M_{\text{out}}} \theta^i_t$), can be used. For parameters taking integer values, like N, this mean posterior can be rounded to the nearest value. Eq 13 thus becomes

$$x^*_{t+1} \approx \underset{x_{t+1}\in\mathcal{S}_{t+1}}{\arg\min} \int dy_{t+1}\, p(y_{t+1}|h_t, x_{t+1}, \hat{\theta}_t) H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = y_{t+1}) \tag{14}$$

Depending on the nature of the studied system and on the time constraints of the experiment, different estimators can also be used, such as e.g. $\hat{\theta}_t = \arg\max_\theta p(\theta|h_t)$. It is important to note that the posterior distribution $p(\theta|h_t)$ is only reduced to a Dirac distribution $\delta(\theta - \theta_t)$ here (in order to obtain a point-based estimate of θ and to simplify the computation of the integral over y_{t+1}), but not for computing the computational entropy $H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = y_{t+1})$ (see below). Secondly, instead of computing the full expectation over the future observation, we set y_{t+1} to its expected value; Eq 13 thus becomes

$$x^*_{t+1} \approx \underset{x_{t+1}\in\mathcal{S}_{t+1}}{\arg\min} H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = \mathbb{E}(Y_{t+1}|h_t, x_{t+1}, \hat{\theta}_t)) \tag{15}$$

In the general case, $\mathbb{E}(Y_{t+1}|h_t, x_{t+1}, \hat{\theta}_t)$ can be computed using Bayesian Quadrature [28]. More specifically, for our model of a chemical synapse, an analytical formulation for the expected value $\mathbb{E}(Y_{t+1}|x_{1:t+1}, \hat{\theta}_t)$ can be efficiently derived using mean-field approximations (see Section Mean-field approximation of vesicle dynamics). For each candidate x_{t+1} in a given finite set $\mathcal{S}_{t+1}$, the entropy $H(\Theta|h_t, x_{t+1}, Y_{t+1} = \mathbb{E}(Y_{t+1}|h_t, x_{t+1}, \hat{\theta}_t))$ can be computed using Algorithm 1.

Finally, for computational efficiency reasons, instead of actually computing the posterior entropy, we are using an upper bound of this posterior entropy at any time step t, i.e. $\frac{1}{2}\log|2\pi e\Sigma_t|$, where Σ_t is the covariance matrix of the particles $\{\theta^i_t\}_{1\leq i\leq M_{\text{out}}}$. Indeed, the maximum entropy distribution for a given covariance matrix Σ_t and a given mean μ_t is precisely the Gaussian distribution $\mathcal{N}(\mu_t, \Sigma_t)$ for which the entropy is $\frac{1}{2}\log|2\pi e\Sigma_t|$. It should be remembered that the in the limit of large amount of data, the posterior distribution will converge to a Gaussian distribution [29]. As a consequence the upper bound will be tight after a large number of observations. Note that this cost function has been previously used in Bayesian Active Learning [10]. Hence, the entropy in 15 is approximately minimized by minimizing the determinant of the covariance matrix of the particles drawn from $p(\Theta|h_t, x_{t+1}, Y_{t+1} = \mathbb{E}(Y_{t+1}|h_t, x_{t+1}, \hat{\theta}_t))$:

$$H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = \mathbb{E}(Y_{t+1}|h_t, x_{t+1}, \hat{\theta}_t)) \approx \frac{1}{2}\log|2\pi e\Sigma_{t+1}| \tag{16}$$

First setting: Reducing the uncertainty of estimates for a given number of observations

From the experimentalist point of view, a highly relevant question is how to optimize the stimulation protocol such that the measured EPSCs are most informative about synaptic parameters. Previous studies showed that some stimulation protocols are more informative than others, but ignored the temporal correlations of the number of readily-releasable vesicles [30] or did not compute which protocol would be most informative [1]. In classical deterministic experiment protocols, the stimulation times $x_{1:T}$ are defined and fixed prior to the recordings. By contrast, active learning optimises the protocol on the fly as data are recorded.

Results for a simulated experiment with ground-truth parameters $N^* = 7$, $p^* = 0.6$, $q^* = 1$ pA, $\sigma^* = 0.2$ pA, and $\tau_D^* = 0.25$s (i.e. the same set of parameters θ^* used in [2]) are displayed in Fig 2A. Here, we compare ESB-BAL to three deterministic protocols:

- in the *Constant* protocol, the synapse is probed at a constant frequency, i.e. $x_t = \text{cst}$;
- in the *Uniform* protocol, ISIs are uniformly drawn from a set $\mathcal{S}$ of candidates x_t consisting of equidistantly separated values ranging from $x^{\min} = 0.005$s (i.e. one order of magnitude shorter than the shortest ISI used in [1]) to $x^{\max}$, i.e. $x_t \sim \text{Uniform}([0.005, x^{\max}])$;
- finally, in the *Exponential* protocol, ISIs are drawn from an exponential distribution with mean τ. Such a protocol has been shown to provide better estimates of synaptic parameters compared to periodic spike trains with constant ISI [1, 30].

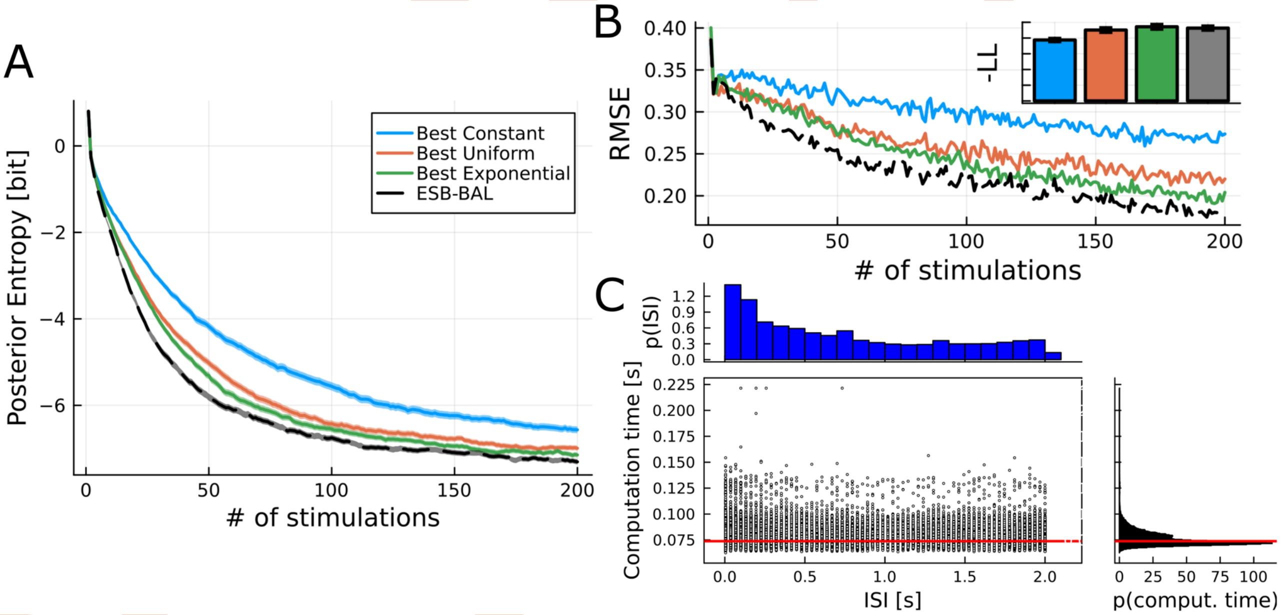

Fig 2. First setting: reducing the uncertainty of estimates for a given number of observations. A: Entropy of the posterior distribution of θ vs. number of observations for different stimulation protocols. Synthetic data were generated from a model of synapse with ground truth parameters $N^* = 7$, $p^* = 0.6$, $q^* = 1$ A, $\sigma^* = 0.2$ A, and $\tau_D^* = 0.25$s [2]. Traces show average over 400 independent repetitions. Shaded area: standard error of the mean. B: RMSE for the same simulations. C: Histograms and scatter plot of the ISIs and the corresponding computation times for the ESB-BAL simulations. Note that the median computation time (horizontal red line) of 74*ms* corresponds to the time required to test 64 candidate intervals: hence, each tested interval takes approx. 1.16*ms*.

https://doi.org/10.1371/journal.pcbi.1011342.g002

The efficiency of these deterministic protocols will depend on their respective parametrizations. To conservatively assess ESB-BAL, we optimize the values of x_t, $x^{\max}$, and τ so that the *Constant*, *Uniform*, and *Exponential* protocols have the best possible performance for the used ground-truth parameters θ^*. S2 Fig. shows the average final entropy decrease (i.e. the information gain) after 200 observations using the *Constant* (top), *Uniform* (middle), or *Exponential* (bottom) protocol, for different values of their hyperparameters. These deterministic protocols (with their optimal respective parametrizations) are then compared to ESB-BAL.

For the different protocols, the average (over 100 independent repetitions) joint differential entropy of the posterior distribution of parameters is plotted as a function of the number of observations (Fig 2A). ESB-BAL allows to reduce the uncertainty (as measured by the entropy) of the parameter estimates for a given number of observations. It should be noted that it is compared to deterministic protocols whose respective hyperparameters have been optimized offline, knowing the value of θ^*. In real physiology experiments, classical protocols are non-adaptative and are defined using (possibly sub-optimal) default parameters. In contrast, in active learning the protocol is optimized on the fly as data are recorded, and its performance will not depend on a prior parametrization. Approximate optimal design via ESB-BAL thus outperforms the best possible *Constant*, *Uniform*, and *Exponential* protocols. Interestingly, it also outperforms a non-parametric optimized design (S3 Fig).

We also verify that ESB-BAL does not lead to biased estimates of θ, as its average RMSE outperforms that of other protocols (Fig 2B). Moreover, for each protocol, we assess whether estimated parameters are a good match to held-out data. An estimate of the parameters $\hat{\theta}_t$ is obtained after $t = 100$ observations, and its likelihood is computed for observations up to $t = 200$. The inset in Fig 2B shows the mean negative log-likelihood of $\hat{\theta}_t$ for 50 repetitions of the protocols in Fig 2A: *Constant* (blue), *Uniform* (orange), *Exponential* (green), and ESB-BAL (grey). Results show no significant differences in the goodness of fit of estimated parameters for held-out data for the *Uniform*, *Exponential*, and ESB-BAL protocols. Interestingly, the *Constant* protocol yields a higher likelihood, but a higher RMSE, for estimated parameters. This means that, for this protocol, estimated parameters are a good fit to data, but that the said data are not informative enough to accurately infer the ground-truth values of the parameters.

Finally, we verify that ESB-BAL is sufficiently fast for online applications, as computation time exceeds the ISI in only a small proportion of cases (Fig 2C). Similar results can be observed for different sets of ground-truth parameters θ^* (S4 Fig) or when only optimizing for the entropy of a specific parameter (S5 Fig).

The computational efficiency of ESB-BAL is achieved through approximations in the computations required to implement the controller. To assess the effect of the sample approximations made in Eqs 14 and 15 on accuracy, we compared ESB-BAL to exact active learning, in which Eq 13 is computed exactly using MC samples (Fig 3). In ESB-BAL (MC θ), samples to compute the expectation over θ are drawn from $p(\theta|h_t)$, and corresponding point-based estimates of y_{t+1} are computed using Eq 23, as in Eq 15. Further, in ESB-BAL (MC θ, y) samples used to compute the expectation over y_{t+1} are drawn by randomly sampling from the inner particles (see Materials and methods). This shows that the approximations used in Algorithm 2 to make active learning online have only a small effect on performance.

Second setting: Reducing the uncertainty of estimates for a given experiment time

Active learning allows, for a given number of observations, to improve the reliability of the estimated parameters. However, in its classical implementation, only the next stimulus input is optimized, disregarding all future observations in the experiment (see the approximation

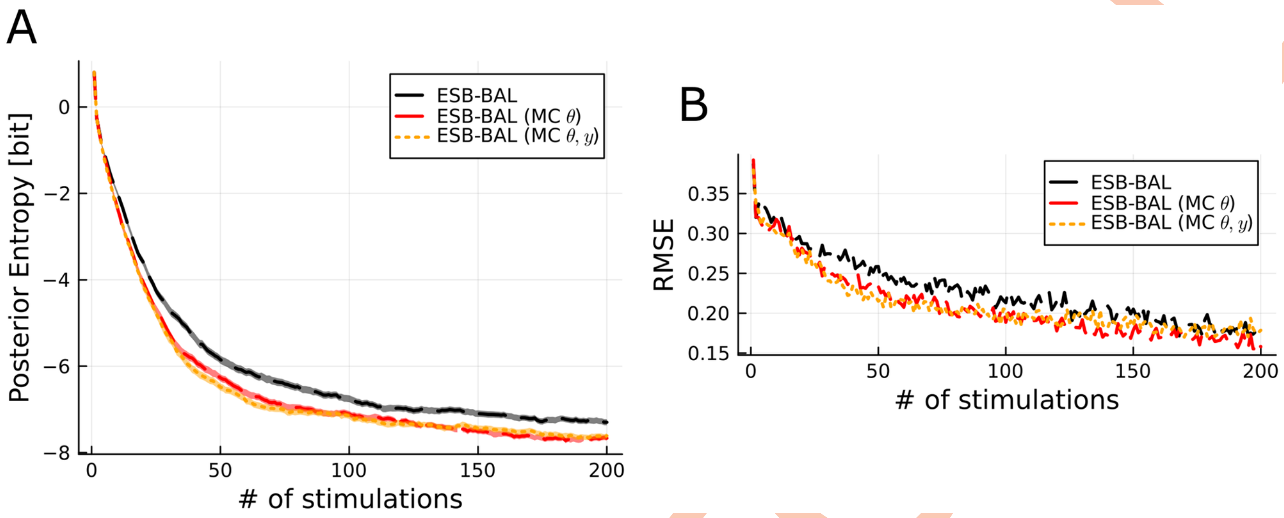

Fig 3. Assessing the effect of point-based approximations on the accuracy of ESB-BAL. Same setting as in Fig 2. In ESB-BAL (MC θ), the integral over θ in Eq 13 is computed using MC samples instead of the point-based approximation described in Eq 14. In ESB-BAL (MC θ, y), both integrals over θ and y_{t+1} in Eq 13 are computed using MC samples instead of the point-based approximations described in Eqs 14 and 15.

https://doi.org/10.1371/journal.pcbi.1011342.g003

made from Eqs 5 to 6). This myopic approach is thus sub-optimal. Moreover, neurophysiology experiments are not only constrained by the number of observations, but also by the total time of the experiment. Since cell viability and recording stability may become limiting during an experiment, the total time of an experimental protocol $x_{\text{tot}} = \sum_{t=1}^{T} x_t$ also needs to be accounted for. Here, to account for the total time of the experiment, and to globally optimize the information gain per unit of time, we can go back to the mutual information expression in Eq 6 and use the chain rule to rewrite it as the following sum:

$$I_{x_{t+1:t+n}}(\Theta; Y_{t+1:t+n}|h_t) = I_{x_{t+1}}(\Theta; Y_{t+1}|h_t) + \sum_{i=2}^{n} I_{x_{t+1:t+i}}(\Theta; Y_{t+i}|Y_{t+1:t+i-1}, h_t) \qquad (17)$$

where the first term on the r.h.s of Eq 17 is the "myopic term" that has been kept in Eq 6 while the second one is the "non-myopic term" and describes the information gain due to all the future events (from $t + 2$ to $t + n$), but still depends on x_{t+1}. Computing this "non-myopic term" is computationally prohibitive. However, instead of simply ignoring it, we can approximate it. If we make the (rather strong) assumption that the future information gain is obtained at a constant rate η (in bits per seconds), then the information gain during the remaining time $x_{\text{tot}} - \sum_{i=1}^{t+1} x_i$ is given by $\eta(x_{\text{tot}} - \sum_{i=1}^{t+1} x_i) = -\eta x_{t+1} + c$ where the constant c is independent of x_{t+1}. With this assumption, we can express the updated formulation of active learning (see original formulation in Eq 13) as

$$x^{*(\eta)}_{t+1} = \underset{x_{t+1}\in\mathcal{S}_{t+1}}{\arg\min}\, H_{x_{t+1}}(\Theta|h_t, Y_{t+1}) + \eta x_{t+1} \qquad (18)$$

where ηx_{t+1} acts as penalty term for longer ISIs. The effect of the assumed future information rate η on the entropy of the posterior distribution of the parameters is displayed in Fig 4A. As expected, adding a penalty term to Eq 13 reduces the precision of the inferred parameter. The loss of information gain increases with the penalty weight η. However, increasing η also increases the speed of information gain, as seen in Fig 4B. Depending on the available time for

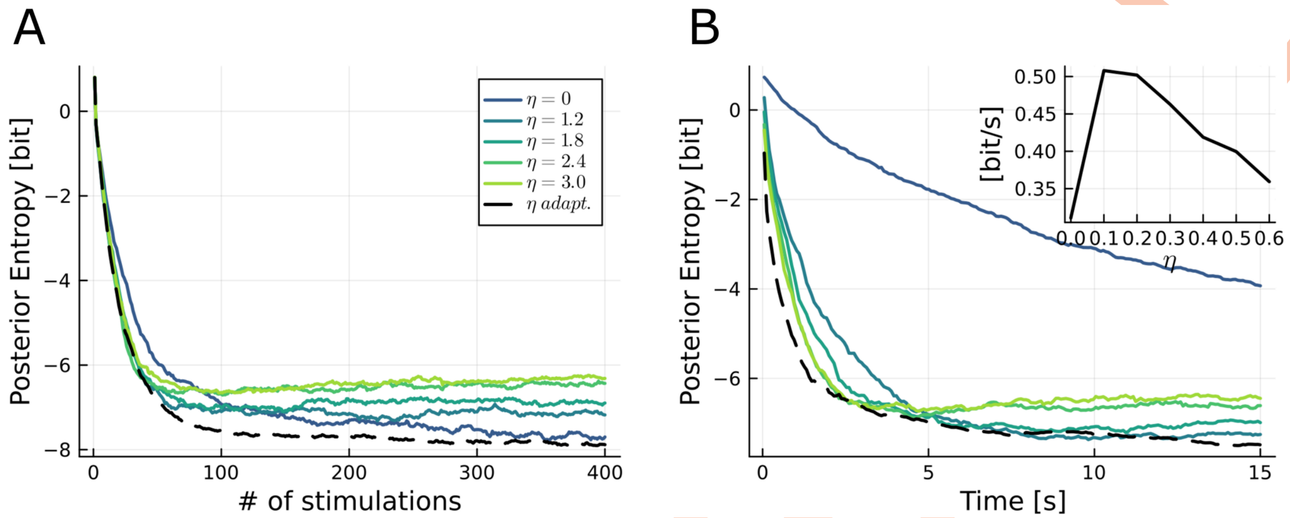

Fig 4. Second setting: reducing the uncertainty of estimates for a given experiment time (effect of penalizing long ISIs on parameter estimates uncertainty and rate of information gain). A: Posterior entropy $H(\Theta|h_t)$ as a function of the stimulation number t for different values of η in Eq 18. Same settings as in Fig 2. B: Same results, but displayed as a function of time. Inset: slope of the entropy vs. time curves (i.e. information rate) vs. η after 10 seconds.

https://doi.org/10.1371/journal.pcbi.1011342.g004

the experiment, it is thus possible to tune η so as to find a trade-off between long-term precision (Fig 4A) and information rate (Fig 4B).

So far, we made the strong assumption that η is constant which is problematic for two reasons. Firstly, it is not accurate (the posterior entropy doesn't decay linearly in time, as seen on Fig 4B) and secondly, it requires to chose η before the start of the experiment. To circumvent both issues at the same time, we implemented an adaptive estimation of the information rate:

$$\eta_{t+1} = \alpha \frac{\Delta H_t}{x_t} + (1 - \alpha)\eta_t \tag{19}$$

where $\Delta H_t = H(\Theta|h_{t-1}) - H(\Theta|h_t)$ is the gain in entropy and α is the learning rate of the estimated information rate η_t. As we can see in Fig 4, the adaptive estimated information rate provides a better performance than a fixed η.

Third setting: Batch optimization and application to neural recordings

To reduce computational complexity, classical implementations of sequential experiment design usually only optimize for the immediate next observation, as in Figs 2, 3, and 4. However, it might be critical for some systems to optimize not only the next stimulus, but rather the next n stimuli of the experiment altogether (see Eq 5) [6, 12]. Synaptic characterization is a telling example: indeed, STD can only be observed for specifically organized batches of stimulation times, where the pool of presynaptic vesicles is first depleted by high-frequency stimulations and its refilling rate then probed using increasing ISIs. Moreover, it should be noted that, depending on the experimental set-up, optimizing batches of future stimuli is often simpler than optimizing each future stimulus. Indeed, some experimental apparatuses (such as the amplifiers used for synaptic stimulation), do not allow for online closed-loop input computation, but only accept programmed batches of inputs. Batch optimization thus allowed to circumvent hardware limitations.

When probing the presynaptic cell, neuroscientists usually use repetitions of a spike train consisting of a tetanic stimulation phase (sustained high-frequency stimulation used to deplete the presynaptic vesicles) followed by recovery spikes at increasing ISIs to probe the STD time constant [31]. These spike trains (especially the duration and frequency of the tetanic phase, and the ISI between recovery spikes) are usually not optimized, and are held constant throughout an entire experiment. Here, we show that ESB-BAL can be used to extend active learning to non-myopic designs (i.e. the optimization is not restricted to the timing of the next event, but takes into account the *n* next input simuli). Such an approach has already been proposed to enable the selection of maximally informative stimulus sequences, hence avoiding the drawbacks of selecting only one stimulus at a time [11].

It should be stressed here that systematically searching over an *n*-dimensional space is computationally prohibitive when *n* is large. As a consequence, we restricted the optimisation to a low-dimensional subspace parametrised by 3 parameters (m, f, x^{last}), see caption of Fig 5A for more details. Candidate batches are defined as such:

- n is the total number of stimulations;
- $m < n$ is the number of spikes in the tetanic stimulation phase;
- f is their frequency;
- x^{last} characterizes the distribution of the recovery spikes (see Materials and methods for details).

A set of candidate batches $\mathcal{S}_{t+1:t+n}$ is thus defined by spanning different values for *m*, *f*, and x^{last}. Algorithm 3, which is a generalization of Algorithm 2, is used to select the next batch of *n* stimuli $x^*_{t+1:t+n}$ in the set $\mathcal{S}_{t+1:t+n}$. Every *n* observations, $H_{x_{t+1:t+n}}(\Theta|h_t, Y_{t+1:t+n})$ is computed using *n* iterations of the filter (i.e. Algorithm 1), in order to pick the optimal next batch $x^*_{t+1:t+n}$ that minimizes the quantity $H_{x_{t+1:t+n}}(\Theta|h_t, Y_{t+1:t+n})$ (i.e. the posterior entropy over the

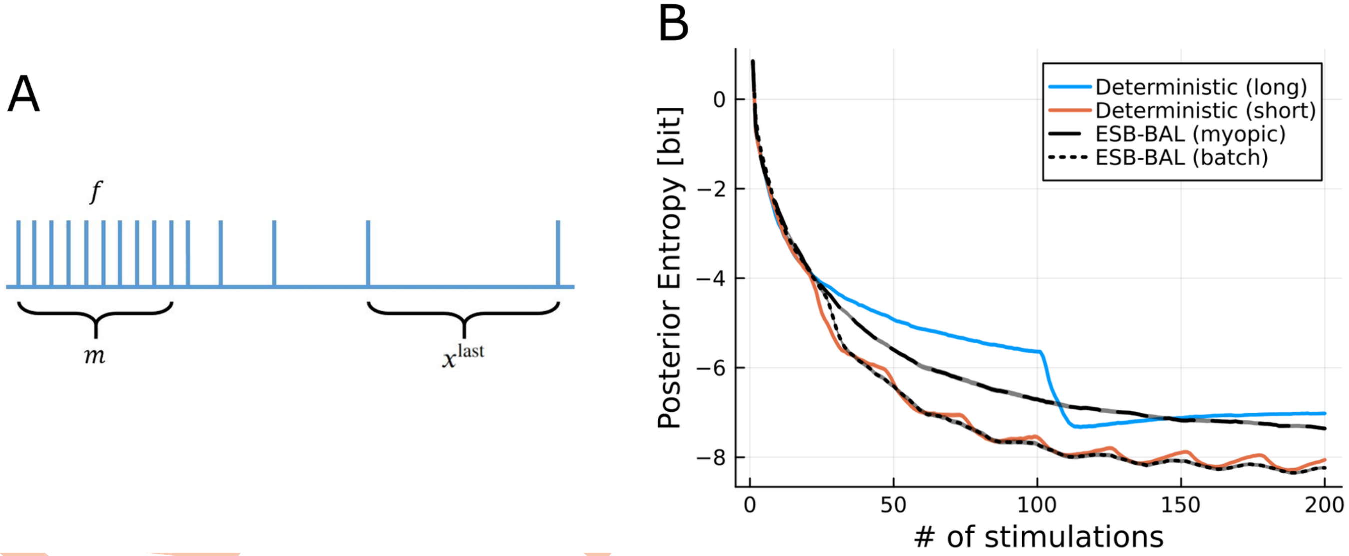

Fig 5. Third setting: batch optimization. A: Schematic of how elements in $\mathcal{S}_{t+1:t+n}$ in Algorithm 3 are defined. They are chosen to span 3 parameters: the number $m < n$ of spikes in the tetanic stimulation phase, the frequency *f* of spikes in the tetanic stimulation phase, and the duration of the final recovery ISI x^{last}. B: Simulated experiment with ground-truth parameters $N^* = 47$, $p^* = 0.27$, $q^* = 2.65$ pA, $\sigma^* = 1.32$ pA, and $\tau_D^* = 0.17$s (i.e. the MAP values from one example cell studied in Fig 6B).

https://doi.org/10.1371/journal.pcbi.1011342.g005

parameters at time step $t + n$ given all observations up to time t):

$$x^*_{t+1:t+n} = \underset{x_{t+1:t+n}\in\mathcal{S}_{t+1:t+n}}{\arg\min}\ H_{x_{t+1:t+n}}(\Theta|h_t, Y_{t+1:t+n}) \tag{20}$$

Fig 5B shows results from a simulated experiment comparing 4 different protocols:

- The *Deterministic (long)* protocol consists of repetitions of a spike trains made of $m = 100$ stimuli at $f = 100Hz$ (tetanic phase) followed by 6 recovery pulses at increasing intervals. Therefore, the long protocol consists of $n = 106$ spikes;
- The *Deterministic (short)* protocol is similar to *Deterministic (long)*, except that its tetanic phase only consists of $m = 20$ pulses instead of 100 (hence consisting of $n = 26$ spikes);
- *ESB-BAL (myopic)* optimizes each stimulus, as in Figs 2, 3, and 4;
- *ESB-BAL (batch)* performs batch optimization, as detailed above: every n observation, the next batch of n stimuli $x^*_{t+1:t+n}$ is computed in the set $\mathcal{S}_{t+1:t+n}$.

Several observations can be made. Firstly, the *Deterministic (long)* protocol is outperformed by its short counterpart, as the latter yields a larger entropy decrease for a given number of stimulations. Intuitively, this highlights that, apart from the first spikes, the tetanic phase (similarly to the *Constant* stimulation from Fig 2) brings little information about the unknown parameters, and that a few high-frequency stimulations followed by recovery spikes are enough to efficiently probe the synapse. Secondly, batch optimization outperforms myopic optimization, showing that synaptic parameter inference benefits from optimizing not only the next stimulus, but rather the next n stimuli of the experiment altogether. Finally, *ESB-BAL (batch)* does not outperform the *Deterministic (short)* protocol, which is likely due to the fact that the latter has been specifically tailored for studying STD and has been defined and optimized through trial and error. Note that the posterior entropy may increase during the tetanic stimulation phase, as explained in Section Particle filtering for synaptic characterization.

We validate our method by applying it to EPSC recordings from mossy fiber to granule cell synaptic connections in acute mouse cerebellar slices (Fig 6A), which depressed nature (S6 Fig) makes them a good match for our theoretical model. Each synapse was stimulated using successively the *Deterministic (long)*, *Deterministic (short)*, and *ESB-BAL (batch)* protocols. For each stimulation protocol, the posterior distribution of the parameters was computed offline using the Metropolis-Hastings algorithm. Fig 6B shows, for different numbers of observations t, the information gain when comparing the *Deterministic (long)* protocol to ESB-BAL (i.e. the entropy after the deterministic protocol minus the entropy after ESB-BAL) across all studied synapses: a positive value signifies a lower entropy when using ESB-BAL. Our experimental results are in line with simulations (Fig 5B), as ESB-BAL outperforms the *Deterministic (long)* protocol at the beginning of the experiment (i.e. when t is low), but not the *Deterministic (short)* protocol (S7 Fig).

Discussion

We developed a method called Efficient Sampling-Based Bayesian Active Learning (ESB-BAL) for approximate optimal experimental design. Using particle filtering, ESB-BAL selects the next experimental design to maximize the approximate mutual information between the output of the experiment and the constants of the studied system. To validate it, we apply ESB-BAL to the problem of estimating the constants of a chemical synapse from its postsynaptic currents evoked by presynaptic stimulations. After each new observation, the optimal next stimulation time can be computed using ESB-BAL. Using synthetic data and synaptic whole-

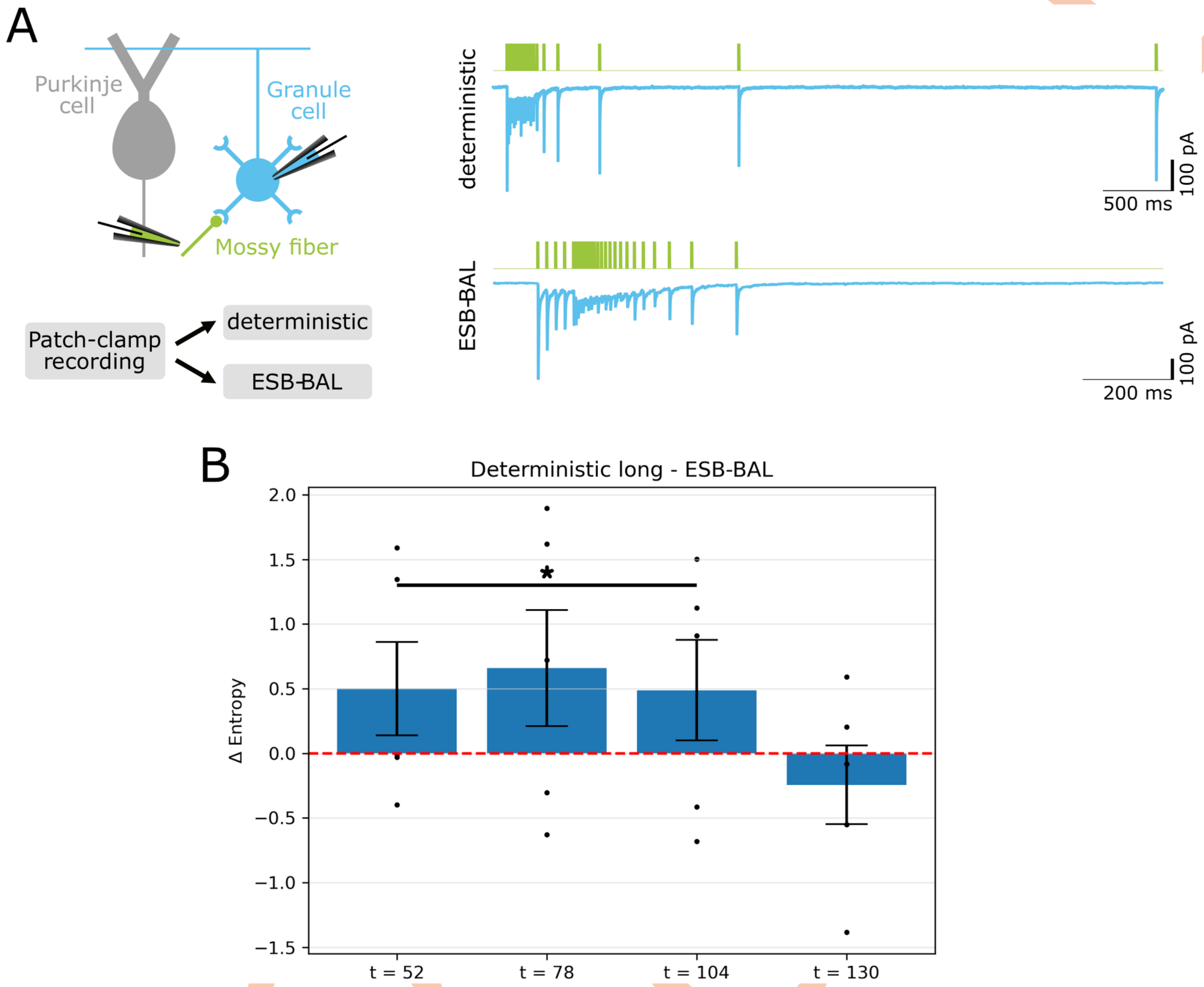

Fig 6. Application to neural recordings. A: Left: Mossy fiber to granule cell synaptic connections from acute cerebellar slices of mice were studied. Each of them was stimulated using successively deterministic protocols and ESB-BAL. Right: examples of postsynaptic current traces recorded from a granule cell upon extracellular mossy fiber stimulation. B: Information gain when comparing the *Deterministic (long)* protocol to ESB-BAL (i.e. the entropy after the deterministic protocol minus the entropy after ESB-BAL) across all studied synapses. A positive value for Δ*Entropy* signifies a lower entropy when using ESB-BAL. Results displayed for different numbers of observations *t*. Test: regression analysis ($p = 0.0381$) comparing entropies after *Deterministic (long)* and ESB-BAL for $t = 52$ to $t = 104$ (see Materials and methods for details).

https://doi.org/10.1371/journal.pcbi.1011342.g006

cell patch-clamp recordings in cerebellar brain slices, we show that our method is efficient and fast enough to be used in real-time biological experiments and can reduce the uncertainty of inferred parameters.

For illustrative purposes, we applied ESB-BAL to the specific problem of estimating the parameters characterizing a chemical synapse. However, we argue that our framework is sufficiently general and efficient to be applicable to a broad range of systems and domains of research. Especially, our extension of the Nested Particle Filter can be applied to any state-space system, even time-variant ones. Moreover, as the Nested Particle Filter is robust to time-varying parameters and model uncertainties [13], we believe that our proposed solution will be especially relevant for neurophysiology experiments or for clinical applications, for instance for optimizing Deep Brain Stimulation (DBS) for the treatment of Parkinson's Disease [32, 33].

We expect active learning to be particularly beneficial to neurophysiology experiments involving live cells or subjects. By reducing the number of samples required to obtain a certain result, or by improving the efficiency of information gain, we can reduce the cost of the experiment and the need for animal subjects. A possible negative impact would be that improving the relative efficiency of neurophysiology experiments may lead to a larger field of applications and therefore a larger demand for animal experiments, analogously to Jevons Paradox [34].

Our approach has some room for improvements. An evident drawback of using particle filtering is that it requires a very large number of particles to provide low variance estimates, as the approximation error only decreases with the square root of the number of particles. More generally, the possibility to efficiently apply particle filtering to any statistical model, irrespective of the dimensions for its state variables and observations, is still an open question, as a general theoretical result linking convergence rate and number of states in a fixed way is lacking. The possibility to apply our ESB-BAL framework to other models and experimental settings should first be verified via simulations. Moreover, future theoretical work should focus on obtaining results on the convergence of the estimators when using active learning. When observations are independent and identically distributed (i.i.d.), active learning will give an unbiased estimate of the parameters, whose variance will decrease with the number of observations [29]. Such theoretical results lack for systems with correlated outputs (such as the EPSCs in the studied synapse model), possibly leading to information saturation [35] or biased estimates.

For experimental applications, the amount of time taken to select a new stimulus is of utmost importance. This is especially true for the system studied here, as the selected next ISI should not be shorter than the time it took to compute it. Hence, an approximate, but fast, filtering algorithm is more useful than an exact filtering method, such as the forward algorithm. The particle nature and the recursive structure of the NPF (which makes it computationally efficient) is enabled by approximating the filtering step (which is valid in the limit of a small variance for the jittering kernel), as described in Section The filter: Online computation of the posterior distributions of parameters. This comes at the cost of an approximation error that decreases more slowly with the number of used particles than for other particle filtering schemes, such as the SMC^2 method [36]. Depending on the constraints of the experiment, this trade-off between computational efficiency and accuracy in the estimate the posterior distribution of the parameters $p(\theta|h_t)$ can be adjusted by implementing a different filtering scheme.

As an application example, we used ESB-BAL to infer the parameters of an idealized model of chemical synapses, which relies on several assumptions and simplifications. Namely, it assumes that the postsynaptic currents elicited by separate vesicle releases add linearly, so that the final current after k_t vesicle openings is qk_t. This assumption disregards possible presynaptic asynchronous release and postsynaptic receptors saturation. There is evidence of linear summation of quanta for the synapses we studied in the experiments [37], but this may not apply to other synapses. Moreover, it also assumes that presynaptic release sites are homogeneous, and share the same values for p, q, and τ_D. Finally, it only accounts for evoked releases and for monosynaptic connections. These assumptions have been widely used in recent models of synaptic transmission, as they allow for tractable analyses while still reflecting the actual observed cellular dynamics [1, 2]. Future experimental work should focus on implementing ESB-BAL for different and more complicated models of a chemical synapse, including for instance short-term facilitation [1, 2, 21, 30, 38] or vesicle content variability [39, 40].

For some recordings in Fig 6B, the benefit of using ESB-BAL instead of a deterministic protocol might seem non-significant. For some synaptic connections (e.g. negative values in Fig

6B), ESB-BAL even yields a higher entropy for the posterior distribution of θ. Different explanations can be put forward. Firstly, it is possible that the classically used deterministic protocols (20 stimuli at 100Hz followed by 6 recovery spikes at increasing ISIs, see Fig 6A) are already well informative about the synaptic parameters. For these protocols, the tetanic stimulation phase and the long inter-sweep interval allow to estimate the value of the hidden states n_t and k_t with a high accuracy, which facilitates the estimation of the synaptic parameters [1]. Moreover, the recovery spikes at varying ISIs are known to be more informative about the synapse's dynamics than a constant stimulation frequency [30]. Secondly, the small benefit of using ESB-BAL for some synaptic connections might be due to a model mismatch issue, as the model defined by Eqs 7 to 10 might not exactly represent the ground-truth mechanisms of the studied synapses. Although mossy fiber to granule cell synapses are believed to be good examples of depressing synapses, our simplified model disregards several aspects of synaptic transmission such as facilitation, postsynaptic saturation, or presynaptic vesicles heterogeneity [37, 41, 42]. These assumptions might explain why ESB-BAL performs more consistently in simulations than in real experiments.

An interesting area of future research would be to formulate Optimal Experiment Design as an optimal control problem, using the framework of the Bellman equation [43, 44]. This multi-stage optimization problem could be solved exactly by defining the associated Bellman equation, in which $I(\Theta;Y_{1:T})$ is the objective function, current observation y_t is the state, input x_t is the control, and where the optimal policy determines the next input x_{t+1}. This approach would allow to account for the remaining available experimental time. Active Learning has been successfully framed as a Deep Reinforcement Learning problem to select which samples should be used to train Natural Language Processing models when the budget for manual annotations is limited [45]. A policy can be learned on a high-resource language (e.g. English) and then used for another language in which annotated training samples are sparse. In biological experiments and clinical settings, whether a policy can be learned on a cell or a subject and applied to another needs to be investigated.

Bayesian Active Learning is an efficient framework for solving the problem of optimal experiment design for parameters inference. Its goal is, for a given generative model $\mathcal{M}$, to optimize the accuracy of the estimates of the parameters θ of $\mathcal{M}$, i.e. to minimize the entropy of the posterior distribution $p(\theta|x_{1:T}, y_{1:T}, \mathcal{M})$. But it is also possible to extend optimal experiment design to model selection: in this setting, the goal is to maximize the discriminability between competing candidate models, i.e. to minimize the entropy of $p(\mathcal{M}|x_{1:T}, y_{1:T})$. Different schemes for OED for model selection have been proposed (see [46] for a discussion), but their computational complexity is a major impediment to their concrete applicability. An interesting future application of ESB-BAL would be to extend it to optimal model selection.

Overall, we expect our proposed solution to pave the way towards better estimates of stochastic models in neuroscience, more efficient training in machine learning, and more systematic and automated experimental designs.

When designing an experiment in physiology, or when training a model on data in machine learning, it is common to choose a priori a fixed set of inputs to the studied system. The use of such non-adaptive, non-optimized protocols often leads to a large variance of the estimated parameters, even when using a large number of trials or data points. Bayesian active learning is an efficient method for optimizing these inputs, but exact solutions are often intractable and not applicable to online experiments. Here, we introduce ESB-BAL, a novel framework combining particle filtering, parallel computing, and mean-field theory. ESB-BAL is general and sufficiently efficient to be applied to a wide range of settings. We use it to infer the parameters of a model of synapse: for this specific example, computation time is a critical constraint, since the typical ISI is shorter than 1s, and because several future inputs need to be optimized together. Using synthetic data and neural recordings, we show that our method has the potential to significantly improve the precision and speed of model-based inferences.

Materials and methods

Ethics statement

Animals were treated following national and institutional guidelines. The Cantonal Veterinary Office of Zurich approved all experiments (authorization no. ZH009/2020).

Particle filtering for synaptic characterization

Initialisation. Computing the posterior distribution of θ firstly implies to specify a prior $p(\theta)$ from which the initial particles $\{\theta_0^i\}_{1\leq i\leq M_{\text{out}}}$ will be drawn. For simplicity, we consider here uniform priors (as in [2, 21]), although the algorithm readily extends to different choices of prior.

Similarly, initial samples for the hidden states $\{n_0^{i,j}, k_0^{i,j}\}_{1\leq j\leq M_{\text{in}}}$ need to be drawn. For $i \in \{1, \ldots, M_{\text{out}}\}, j \in \{1, \ldots, M_{\text{in}}\}$, we define:

- $n_0^{i,j} = N_i$ (i.e. all vesicles are supposed to be in the readily-releasable state at the beginning of the simulation);
- $k_0^{i,j} \sim \text{Bin}(N_i, p_i)$ (consistently with Eq 9).

Jittering step. The parameters that we wish to infer are supposed to be constant. It is thus impossible to define dynamics of the form $p(\theta_{t+1}^i|\theta_t^i)$ for the particles (as opposed to filtering problems aiming at inferring a dynamical hidden state, as for instance in [47]). To avoid particle degeneracy, it is thus necessary to mutate particles using a jittering kernel $\kappa(\theta_{t-1}^i)$. When particles take continuous values, a classical choice for the jittering kernel is to draw the next particle θ_t^i from a Gaussian distribution with mean θ_{t-1}^i and which variance is called the jittering width (see [13] for a detailed discussion). In our implementation, the range of possible values for each parameter is discretized, so that each particle corresponds to a position on the grid of possible parameters values (same implementation as in [2]). The free parameter α in our jittering kernel thus corresponds to the probability of moving by one bin:

$$\theta_t^i = \kappa(\theta_{t-1}^i) = \begin{cases} \theta_{t-1}^i, & \text{with probability } 1-\alpha \\ \tilde{\theta}_{t-1}^i, & \text{with probability } \alpha \end{cases} \tag{21}$$

where $\tilde{\theta}_{t-1}^i$ is one (randomly chosen) bin away from θ_{t-1}^i.

At each time step, the jittering step tends to increase the entropy of the distribution of the particles (by mutating them). On the other hand, the resampling step (see below) tends to reduce it, by keeping only high-likelihood particles. If the variance of the jittering kernel is sufficiently high to outperform the effect of resampling, the overall entropy of the distribution of the particles will increase (as, for instance, during the tetanic stimulation phase in Fig 5B.

Propagation step. Inner particles are redrawn based on $n_t^{i,j} \sim p(n_t^{i,j}|n_{t-1}^{i,j}, k_{t-1}^{i,j}, \theta_t^i, x_t)$ (Eq 10) and $k_t^{i,j} \sim p(k_t^{i,j}|n_t^{i,j}, \theta_t^i)$ (Eq 9).

Likelihood computation step. $p(y_t|n_t^{i,j}, k_t^{i,j}, \theta_t^i)$ is computed according to Eq 8.

Resampling step. Particles are resampled by multinomial resampling using the algorithm introduced in [48], which allows to draw a list of sorted numbers in a single step. Alternative resampling schemes can also be implemented. For instance, residual and stratified resampling methods dominate the multinomial one in terms of conditional variance [49]. We found that for artificial data, the stratified method leads to faster convergence of the estimates of N, q, and σ, whereas p and τ_D estimates do not show a significant difference of rate of convergence (S8B Fig), but does not significantly improve the information gain (S8A Fig).

Algorithm 2: Computation of the optimal next stimulation time for synaptic characterization

```
set $\hat{\theta}_t = \frac{1}{M_{\text{out}}}\sum_{i=1}^{M_{\text{out}}} \theta_t^i$ (mean from the current posterior estimation);
Input: $\mathcal{S}_{t+1}$ (set of candidates $x_{t+1}$);
for $x_{t+1}$ in $\mathcal{S}_{t+1}$ do
    Compute $\mathbb{E}(Y_{t+1}|x_{1:t+1}, \hat{\theta}_t)$ using Eq 23;
    Compute $p(\Theta|h_t, x_{t+1}, Y_{t+1} = \mathbb{E}(Y_{t+1}|x_{1:t+1}, \hat{\theta}_t))$ using Algorithm 1;
    Compute $H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = \mathbb{E}(Y_{t+1}|x_{1:t+1}, \hat{\theta}_t))$ using Eq 16;
end
$x_{t+1}^* = \text{arg min}_{x_{t+1} \in \mathcal{S}_{t+1}} H_{x_{t+1}}(\Theta|h_t, Y_{t+1} = \mathbb{E}(Y_{t+1}|x_{1:t+1}, \hat{\theta}_t))$
```

Algorithm 2 is slightly modified in Fig 3 for the "ESB-BAL (exact)" simulations, in which Eq 13 is computed using MC samples instead of the point-based simplifications explained in Eqs 14 and 15. Samples to compute the expectation over θ are drawn from the current posterior distribution $p(\theta|h_t)$, i.e. by random sampling from the pool of particles $\{\theta_t^i\}_{i \in \{1,\ldots,M_{out}\}}$. For each of these samples, and for each candidate next input x_{t+1} in $\mathcal{S}_{t+1}$:

- In ESB-BAL (MC θ): the corresponding value of y_{t+1} is computed using Eq 23, as in Eq 15;
- In ESB-BAL (MC θ, y): samples used to compute the expectation over y_{t+1} are drawn by randomly sampling $n_t^{i,j} \sim p(n_t^{i,j}|n_{t-1}^{i,j}, k_{t-1}^{i,j}, \theta_t^i, x_t)$ and $k_t^{i,j} \sim p(k_t^{i,j}|n_t^{i,j}, \theta_t^i)$, and using Eq 8.

Unless otherwise specified, all the simulations results were obtained with $M_{out} = 1024$ and $M_{in} = 256$ particles, and were run using a commercially available Nvidia GTX 1080 Ti GPU.

Input-Output Hidden Markov Model (IO-HMM) of synaptic transmission

In line with previous works (and especially with [1, 2]), we used a simplified Input-Output Hidden Markov Model (IO-HMM) to describe the synapse (as explained in Section The system: A binomial model of neurotransmitter release). Its specificity, compared to classical HMMs, is that N is considered as a parameter to be optimized together with the other ones. Besides, the value of N characterizes the range of values that the hidden states can have, as $0 \leq k_t \leq n_t \leq N$. De facto, the exact value of N for the studied synapse is unknown, can take a broad range of possible values [1], and needs to be inferred from the observations. Different approaches have been used in previous works:

- In [1], the authors fixed the value for N and then estimated the other parameters using the Expectation-Maximization (EM) algorithm. The procedure is repeated for different values of N ranging from 1 to 100, and the set of parameters yielding the highest likelihood is selected as the maximum likelihood estimator. In our setting, this would be equivalent to running several instances of ESB-BAL in parallel for different values of N, which would significantly increase the computational load.
- In [2], the authors used the Metropolis-Hastings algorithm to compute the posterior distribution of the parameters, including N. In our case, we also jointly optimize all parameters, including N. This means that, due to the jittering kernel, N_t^i might change from time to time. As a consequence, we introduce the ReLU function to keep the value of $n_t^{i,j}$ positive in Algorithm 1 (where N_t^i and p_t^i refer to the values of N and p in particle θ_t^i):

$$\begin{aligned}
n_t^{i,j} &\leftarrow ReLU(N_t^i - n_{t-1}^{i,j} + k_{t-1}^{i,j})\\
k^{i,j} &\leftarrow rand(\mathrm{Bin}(n_t^{i,j}, \pi(x_t))\\
n_t^{i,j} &\leftarrow N_t^i - n_t^{i,j} + k^{i,j}\\
k_t^{i,j} &\leftarrow rand(\mathrm{Bin}(n_t^{i,j}, p_t^i))
\end{aligned}$$

Mean-field approximation of vesicle dynamics

Our synapse model, as defined by Eqs 7 to 10, is a Hidden Markov Model with observations y_t and hidden states $z_t = (n_t, k_t)$. The predictive distribution $p(y_{t+1}|h_t, x_{t+1}, \theta)$ used in Eq 1 can be computed using the forward algorithm: however, the algorithmic complexity of this exact filtering procedure, which scales with N^4, makes it impractical for closed-loop applications. Here, we suggest that computation can be massively simplified by using a mean-field approximation of vesicle dynamics: the analytical mean and variance of hidden and observed variables can be computed using recursive formulæ [46].

Let $r_t \in [0, 1]$ denote the average fraction of release-competent vesicles at the moment of spike t. Its values, given $\theta = [N, p, q, \sigma, \tau_D]$ and $x_{1:t}$, can be iteratively computed (see [1], Eq (7)) from the equations of the Tsodyks-Markram model [20]:

$$r_t = 1 - (1 - (1 - p)r_{t-1})\exp\left(-\frac{x_t}{\tau_D}\right) \qquad (22)$$

with $r_1 = 1$. It follows that the expected value of the EPSC after spike t is

$$\mathbb{E}(Y_t|x_{1:t}, \theta) = r_t Npq \qquad (23)$$

One can note that the variance of the number of available vesicles n_t conditioned on the history of previous activations $x_{1:t}$ and on the parameter values θ can be computed similarly using the law of total variance:

$$\mathrm{Var}(n_t|x_{1:t}, \theta) = \mathbb{E}(\mathrm{Var}(n_t|n_{t-1}, k_{t-1}, x_{1:t}, \theta)) + \mathrm{Var}(\mathbb{E}(n_t|n_{t-1}, k_{t-1}, x_{1:t}, \theta)) \qquad (24)$$

Since $n_t = n_{t-1} - k_{t-1} + v_t$ with $v_t \sim \mathrm{Bin}(N - n_{t-1} + k_{t-1}, \pi(x_t))$ (see Eq 10), it follows that

$$\begin{aligned}
\mathrm{Var}(n_t|x_{1:t}, \theta) = \pi(x_t)(1 - \pi(x_t))N(1 - r_{t-1} + pr_{t-1}) +&\\
(1 - \pi(x_t))^2\mathrm{Var}(n_{t-1} - k_{t-1}|x_{1:t-1}, \theta)&
\end{aligned} \qquad (25)$$

Finally, by noting that $(n_t - k_t)|n_t \sim \mathrm{Bin}(n_t, 1 - p)$ and using again the law of total variance to compute

$$\begin{aligned}
\mathrm{Var}(n_{t-1} - k_{t-1}|x_{1:t-1}, \theta) = \mathbb{E}(\mathrm{Var}(n_{t-1} - k_{t-1}|n_{t-1}, x_{1:t-1}, \theta)) +&\\
\mathrm{Var}(\mathbb{E}(n_{t-1} - k_{t-1}|n_{t-1}, x_{1:t-1}, \theta))&
\end{aligned} \qquad (26)$$

we obtain

$$\mathrm{Var}(Y_t|x_{1:t}, \theta) = \sigma^2 + q^2(Nr_t p(1 - p) + \mathrm{Var}(n_{t-1}|x_{1:t-1}, \theta)p^2) \qquad (27)$$

Eqs 23 and 27 are respectively used to compute the expected value of y_t given θ and $x_{1:t}$ (which is used in the point-based approximation of Eq 15) and its variance (which is used to verify the goodness of fit of our model to recorded EPSCs in S6 Fig).

Batch optimization

Each candidate batch of n stimulation times in $\mathcal{S}_{t+1:t+n}$ (Fig 5A) is described by 3 parameters:

- $m < n$: the number of tetanic stimulations [-];
- f: the frequency of the tetanic stimulations [Hz];
- x^{last}: the time interval before the final recovery spike [s].

A train of n stimulations is thus composed of m tetanic stimulations at a frequency f, followed by $n - m$ recovery spikes with increasing inter-spike intervals $\frac{x^{\text{last}}}{n-m}, \frac{x^{\text{last}}}{n-m-1}, \ldots, \frac{x^{\text{last}}}{2}, x^{\text{last}}$. The following values were used during experiments (Fig 6): $n = 26$, $m \in [5, 10, 15, 20]$, $f \in [25Hz, 50Hz, 100Hz, 200Hz]$, $x^{\text{last}} \in [0.1s, 0.5s, 1.0s, 2.0s]$.

These 4 possible values for each parameter m, f, and x^{last} yield 64 different combinations: $\mathcal{S}_{t+1:t+n}$ thus consists of 64 different candidate batches, from which Algorithm 3 picks the next optimal one.

Algorithm 3: Computation of the optimal next batch of ISIs

```
Input: $\mathcal{S}_{t+1:t+n}$ (set of candidates $x_{t+1:t+n}$);
for $x_{t+1:t+n}$ in $\mathcal{S}_{t+1:t+n}$ do
  Compute $H_{x_{t+1:t+n}}(\Theta|h_t, Y_{t+1:t+n})$ using Algorithm 1;
end
$x^*_{t+1:t+n} = \arg\min_{x_{t+1:t+n} \in \mathcal{S}_{t+1:t+n}} H_{x_{t+1:t+n}}(\Theta|h_t, Y_{t+1:t+n})$
```

Electrophysiological recordings

Experiments were performed in adult (> 1-month-old) male and female C57BL/6J mice (Janvier Labs, France). Animals were housed in groups of 3–5 in standard cages on a 12h-light/12h-dark cycle with food and water ad libitum. Mice were sacrificed by rapid decapitation after isoflurane anesthesia. The cerebellar vermis was removed quickly and mounted in a chamber filled with cooled extracellular solution. 300-μm thick parasagittal slices were cut using a Leica VT1200S vibratome (Leica Microsystems, Germany), transferred to an incubation chamber at 35 ˚C for 30 minutes, and then stored at room temperature until experiments.

The extracellular solution (artificial cerebrospinal fluid, ACSF) for slice cutting and storage contained (in mM): 125 NaCl, 25 NaHCO3, 20 D-glucose, 2.5 KCl, 2 CaCl2, 1.25 NaH2PO4, 1 MgCl2, bubbled with 95% O2 and 5% CO2. Slices were visualized using an upright microscope with a 60×, 1 NA water-immersion objective, infrared optics, and differential interference contrast (Scientifica, UK). The recording chamber was continuously perfused with ACSF supplemented with 10 μM D-APV, 10 μM bicuculline, and 1 μM strychnine. Experiments were performed at room temperature (21–25 ˚C). Patch pipettes (open-tip resistances of 3–8 MΩ) were filled with solution containing (in mM): 150 K-D-gluconate, 10 NaCl, 10 HEPES, 3 MgATP, 0.3 NaGTP, 0.05 ethyleneglycol-bis(2-aminoethylether)-N,N,N',N'-tetraacetic acid (EGTA), pH adjusted to 7.3 using KOH.

Voltage-clamp recordings were done using a HEKA EPC10 amplifier controlled via Patchmaster software (HEKA Elektronik GmbH, Germany) essentially as described in [50]. Voltages were corrected for a liquid junction potential of +13 mV. Extracellular mossy fiber stimulation was performed using square voltage pulses (duration, 150 μs) generated by a stimulus isolation unit (ISO-STIM 01B, NPI) and applied through an ACSF-filled pipette. The pipette was moved over the slice surface close to the postsynaptic cell while applying voltage pulses until excitatory postsynaptic currents (EPSCs) could be evoked reliably. Care was taken to stimulate single mossy fiber inputs. EPSCs were recorded at a holding potential of –80 mV; data were low-pass filtered at 2.9 kHz and digitized at 20–50 kHz. Train stimulation protocols comprised bouts of 20 or 100 MF stimulations at 100 Hz, followed by single pulses to monitor recovery from short-term depression (intervals: 25 ms, 50 ms, 100 ms, 300 ms, 1 s, 3 s). The interval between subsequent train recordings was at least 30 s. For OED experiments, custom protocols were generated online as file templates for use with Patchmaster. EPSCs were quantified as peak amplitudes from a 300-µs baseline before onset.

To facilitate the definition of the range of possible values for parameters q and σ (and especially to avoid running an experiment with too narrow ranges), recorded EPSC amplitudes were normalized by dividing them by their maximum value. Analyses are thus performed by assuming $q \in [0, 1]$ and $\sigma \in [0, 1]$.

Data were collected from 12 synaptic connections, each of them being successively stimulated with the *Deterministic (long)*, *Deterministic (short)*, and ESB-BAL protocols. To accurately compare the effect of ESB-BAL to this of deterministic protocols, it is important to ensure that the synapse (and especially the strength of the synaptic connection) has not changed between successive protocols (as synaptic strength can rapidly change in between stimulation protocols due to synaptic plasticity). For each protocol, the mean value of the first EPSC of each batch is computed, and the relative difference of this value for a pair of protocols is used to assess the drift between these protocols. For a given synapse, pairs of protocols (i.e. *Deterministic (long)* vs. ESB-BAL and *Deterministic (short)* vs. ESB-BAL) for which this drift exceeded 10% were discarded, hence effectively leaving 5 pairs of protocols in Fig 6B and 7 in S7 Fig.

EPSC quantification

Data were sampled at 20.000Hz. To quantify EPSC peak amplitudes, we first used a boxcar smoothing of raw data with a 5-point (i.e. 250μs) window. EPSC peak amplitude was then defined as the difference between the mean of a 0.4-ms baseline after the stimulation artifact and the minimum in the following 4-ms time window. Due to the synaptic delay, the EPSC starts about 800μs to 1ms after the stimulation, thus allowing to compute the baseline between the stimulation artifact and the EPSC onset. This has advantages in high-frequency stimulation trains, when the preceding EPSC may not have decayed to baseline.

Statistical testing

Linear regression is used to measure the effect of ESB-BAL compared to a deterministic protocol. For data shown in Fig 6B, entropy (dependent variable) is regressed against the categorical variable corresponding to the used protocol (ESB-BAL or deterministic). t-test is then performed on the fitted slope coefficient.

Supporting information

S1 Fig. Examples of posteriors obtained using the filter (Algorithm 1). Upper left panel: train of synthetic EPSCs generated from the model described in Section The system: A binomial model of neurotransmitter release. Other panels: posterior distributions of the parameters after 230 stimulations. Ground-truth values used to generate the EPSCs are displayed as red vertical lines.
(TIFF)

S2 Fig. Average final entropy decrease (i.e. information gain) after 200 observations using the *Constant* (top), *Uniform* (middle), or *Exponential* (bottom) protocol, for different values of their hyperparameters. Ground truth parameters used are $N^* = 7$, $p^* = 0.6$, $q^* = 1$ A, $\sigma^* = 0.2$ A, and $\tau_D^* = 0.25$s [2]. Vertical red lines indicate the ground truth value $\tau_D^* = 0.25$s used for simulations. Optimal values for x_t, $x^{\max}$, and τ are used in Fig 2. For the hyperparameter x^{max}, values up to 2s are spanned to ensure that the mean of the *Uniform* distribution will span values from 0 to 1s.
(TIFF)

S3 Fig. Comparison of ESB-BAL to an optimized non-parametric fixed design. In previous analyses, our adaptative design (ESB-BAL) was compared to parametric fixed designs (Constant, Uniform, and Exponential). Although intuitive, these parametric fixed protocols may not accurately represent the most informative fixed design. The best fixed ISI distribution can be computed non-parametrically by randomly drawing ISIs from the list of previous ISIs computed via ESB-BAL. We thus obtain a fixed and non-parametric optimized design. Results show that this best fixed design (purple) performs similarly to the best exponential design (green) and is still outperformed by ESB-BAL (black).
(TIFF)

S4 Fig. Same setting as in Fig 2 but for ground truth parameters $N^* = 10$, $p^* = 0.85$, $q^* = 1$ A, $\sigma^* = 0.2$ A, and $\tau_D^* = 0.2$s.
(TIFF)

S5 Fig. Same setting as in Fig 2 but when optimizing solely for the marginal posterior distribution of τ_D.
(TIFF)

S6 Fig. The binomial model provides a good fit to recorded EPSCs. Postsynaptic responses to presynaptic stimulations were recorded in mossy fiber to granule cell synaptic connections from acute cerebellar slices of mice. EPSCs amplitudes are computed from raw traces, as detailed in Materials and methods. In this trace example, the presynaptic axon was stimulated using repetitions of a deterministic train of spikes composed of 20 stimulations at 100Hz (tetanic stimulation) followed by 6 recovery spikes at increasing ISIs. This trace illustrates the short-term depression of the studied synapses, visible in the lower EPSCs amplitudes following the first spike in the tetanic phase, and in the increasing amplitudes during the recovery phase. The goodness of fit of the binomial model (described in Section The system: A binomial model of neurotransmitter release) is assessed by comparing its prediction to recorded EPSCs. For a given synapse, we first obtain maximum a-posteriori estimates of its parameters θ using Metropolis-Hastings samples (as in [2]). Second, at each time step t, the value of the expected EPSC y_t and its variance given θ and $x_{1:t}$ can be computed using Eqs 23 and 27. This prediction from the model (mean: orange solid line; shaded area: 3 standard deviations) can then be compared to actual recordings (blue solid line).
(TIF)

S7 Fig. Same setting as in Fig 6B but comparing *Deterministic (short)* and *ESB-BAL (batch)*.
(TIF)

S8 Fig. Comparison of the effect of Multinomial and Stratified resampling. Same setting as in Fig 2. Simulations used either the Multinomial or Stratified schemes for particles resampling (see Section Particle filtering for synaptic characterization). Although the Stratified resampling scheme improves the convergence of the parameters (B), it does not significantly improve the information gain (A).
(TIFF)

Acknowledgments

Calculations were performed on UBELIX (http://www.id.unibe.ch/hpc), the HPC cluster at the University of Bern. The CUDA.jl package [14, 15] is licensed under the MIT “Expat” License (https://github.com/JuliaGPU/CUDA.jl/blob/master/LICENSE.md).

We thank Ehsan Abedi, Jakob Jordan, and Anna Kutschireiter for the fruitful discussions.

Author Contributions

Conceptualization: Jean-Pascal Pfister.

Formal analysis: Camille Gontier.

Funding acquisition: Jean-Pascal Pfister.

Investigation: Camille Gontier, Igor Delvendahl.

Resources: Martin Müller.

Software: Camille Gontier, Simone Carlo Surace.

Supervision: Martin Müller, Jean-Pascal Pfister.

Visualization: Camille Gontier, Igor Delvendahl.

Writing – original draft: Camille Gontier.

Writing – review & editing: Simone Carlo Surace, Igor Delvendahl, Martin Müller, Jean-Pascal Pfister.

References

1. Barri A, Wang Y, Hansel D, Mongillo G. Quantifying repetitive transmission at chemical synapses: a generative-model approach. Eneuro. 2016; 3(2). https://doi.org/10.1523/ENEURO.0113-15.2016
2. Bird AD, Wall MJ, Richardson MJ. Bayesian inference of synaptic quantal parameters from correlated vesicle release. Frontiers in computational neuroscience. 2016; 10:116. https://doi.org/10.3389/fncom.2016.00116 PMID: 27932970
3. Flesch T, Balaguer J, Dekker R, Nili H, Summerfield C. Comparing continual task learning in minds and machines. Proceedings of the National Academy of Sciences. 2018; 115(44):E10313–E10322. https://doi.org/10.1073/pnas.1800755115 PMID: 30322916
4. Emery AF, Nenarokomov AV. Optimal experiment design. Measurement Science and Technology. 1998; 9(6):864. https://doi.org/10.1088/0957-0233/9/6/003
5. Sebastiani P, Wynn HP. Maximum entropy sampling and optimal Bayesian experimental design. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2000; 62(1):145–157. https://doi.org/10.1111/1467-9868.00225
6. Ryan EG, Drovandi CC, McGree JM, Pettitt AN. A review of modern computational algorithms for Bayesian optimal design. International Statistical Review. 2016; 84(1):128–154. https://doi.org/10.1111/insr.12107
7. Lewi J, Butera R, Paninski L. Sequential optimal design of neurophysiology experiments. Neural computation. 2009; 21(3):619–687. https://doi.org/10.1162/neco.2008.08-07-594 PMID: 18928364
8. Park M, Horwitz G, Pillow JW. Active learning of neural response functions with Gaussian processes. In: NIPS. Citeseer; 2011. p. 2043–2051.
9. Park M, Pillow J. Bayesian active learning with localized priors for fast receptive field characterization. Advances in neural information processing systems. 2012; 25:2348–2356.
10. Jha A, Ashwood ZC, Pillow JW. Bayesian Active Learning for Discrete Latent Variable Models. arXiv preprint arXiv:220213426. 2022;.
11. Lewi J, Schneider DM, Woolley SM, Paninski L. Automating the design of informative sequences of sensory stimuli. Journal of computational neuroscience. 2011; 30:181–200. https://doi.org/10.1007/s10827-010-0248-1 PMID: 20556641
12. Drovandi CC, Tran MN, et al. Improving the efficiency of fully Bayesian optimal design of experiments using randomised quasi-Monte Carlo. Bayesian Analysis. 2018; 13(1):139–162. https://doi.org/10.1214/16-BA1045
13. Crisan D, Miguez J, et al. Nested particle filters for online parameter estimation in discrete-time state-space Markov models. Bernoulli. 2018; 24(4A):3039–3086. https://doi.org/10.3150/17-BEJ954
14. Besard T, Foket C, De Sutter B. Effective Extensible Programming: Unleashing Julia on GPUs. IEEE Transactions on Parallel and Distributed Systems. 2018;.

15. Besard T, Churavy V, Edelman A, De Sutter B. Rapid software prototyping for heterogeneous and distributed platforms. Advances in Engineering Software. 2019; 132:29–46. https://doi.org/10.1016/j.advengsoft.2019.02.002

16. Huan X, Marzouk YM. Simulation-based optimal Bayesian experimental design for nonlinear systems. Journal of Computational Physics. 2013; 232(1):288–317. https://doi.org/10.1016/j.jcp.2012.08.013

17. Foster A, Jankowiak M, Bingham E, Horsfall P, Teh YW, Rainforth T, et al. Variational bayesian optimal experimental design. arXiv preprint arXiv:190305480. 2019;.

18. Del Castillo J, Katz B. Quantal components of the end-plate potential. The Journal of physiology. 1954; 124(3):560–573. https://doi.org/10.1113/jphysiol.1954.sp005129 PMID: 13175199

19. Katz B. The release of neural transmitter substances. Liverpool University Press. 1969; p. 5–39.

20. Tsodyks M, Pawelzik K, Markram H. Neural networks with dynamic synapses. Neural computation. 1998; 10(4):821–835. https://doi.org/10.1162/089976698300017502 PMID: 9573407

21. Gontier C, Pfister JP. Identifiability of a Binomial Synapse. Frontiers in computational neuroscience. 2020; 14:86. https://doi.org/10.3389/fncom.2020.558477 PMID: 33117139

22. Wieland FG, Hauber AL, Rosenblatt M, Tönsing C, Timmer J. On structural and practical identifiability. Current Opinion in Systems Biology. 2021;. https://doi.org/10.1016/j.coisb.2021.03.005

23. Kaeser PS, Regehr WG. The readily releasable pool of synaptic vesicles. Current opinion in neurobiology. 2017; 43:63–70. https://doi.org/10.1016/j.conb.2016.12.012 PMID: 28103533

24. Stricker C, Redman SJ. Quantal analysis based on density estimation. Journal of neuroscience methods. 2003; 130(2):159–171. https://doi.org/10.1016/j.jneumeth.2003.09.021 PMID: 14667544

25. Scheuss V, Neher E. Estimating synaptic parameters from mean, variance, and covariance in trains of synaptic responses. Biophysical journal. 2001; 81(4):1970–1989. https://doi.org/10.1016/S0006-3495(01)75848-1 PMID: 11566771

26. Bykowska O, Gontier C, Sax AL, Jia DW, Montero ML, Bird AD, et al. Model-based inference of synaptic transmission. Frontiers in synaptic neuroscience. 2019; 11:21. https://doi.org/10.3389/fnsyn.2019.00021 PMID: 31481887

27. Kutschireiter A, Surace SC, Pfister JP. The Hitchhiker's guide to nonlinear filtering. Journal of Mathematical Psychology. 2020; 94:102307. https://doi.org/10.1016/j.jmp.2019.102307

28. Acerbi L. Variational bayesian monte carlo. Advances in Neural Information Processing Systems. 2018; 31.

29. Paninski L. Asymptotic theory of information-theoretic experimental design. Neural Computation. 2005; 17(7):1480–1507. https://doi.org/10.1162/0899766053723032 PMID: 15901405

30. Costa RP, Sjostrom PJ, Van Rossum MC. Probabilistic inference of short-term synaptic plasticity in neocortical microcircuits. Frontiers in computational neuroscience. 2013; 7:75. https://doi.org/10.3389/fncom.2013.00075 PMID: 23761760

31. Markram H, Pikus D, Gupta A, Tsodyks M. Potential for multiple mechanisms, phenomena and algorithms for synaptic plasticity at single synapses. Neuropharmacology. 1998; 37(4-5):489–500. https://doi.org/10.1016/S0028-3908(98)00049-5 PMID: 9704990

32. Tinkhauser G, Moraud EM. Controlling Clinical States Governed by Different Temporal Dynamics With Closed-Loop Deep Brain Stimulation: A Principled Framework. Frontiers in neuroscience. 2021; 15. https://doi.org/10.3389/fnins.2021.734186 PMID: 34858126

33. Carè M, Averna A, Barban F, Semprini M, De Michieli L, Nudo RJ, et al. The impact of closed-loop intracortical stimulation on neural activity in brain-injured, anesthetized animals. Bioelectronic Medicine. 2022; 8(1):1–14.

34. Jevons WS. The coal question. An Inquiry Concerning the Prog. 1862;.

35. Moreno-Bote R, Beck J, Kanitscheider I, Pitkow X, Latham P, Pouget A. Information-limiting correlations. Nature neuroscience. 2014; 17(10):1410–1417. https://doi.org/10.1038/nn.3807 PMID: 25195105

36. Chopin N, Jacob PE, Papaspiliopoulos O. SMC2: an efficient algorithm for sequential analysis of state space models. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2013; 75(3):397–426. https://doi.org/10.1111/j.1467-9868.2012.01046.x

37. Sargent PB, Saviane C, Nielsen TA, DiGregorio DA, Silver RA. Rapid vesicular release, quantal variability, and spillover contribute to the precision and reliability of transmission at a glomerular synapse. Journal of Neuroscience. 2005; 25(36):8173–8187. https://doi.org/10.1523/JNEUROSCI.2051-05.2005 PMID: 16148225

38. Pfister JP, Dayan P, Lengyel M. Synapses with short-term plasticity are optimal estimators of presynaptic membrane potentials. Nature neuroscience. 2010; 13(10):1271–1275. https://doi.org/10.1038/nn.2640 PMID: 20852625

39. Bhumbra GS, Beato M. Reliable evaluation of the quantal determinants of synaptic efficacy using Bayesian analysis. Journal of neurophysiology. 2013; 109(2):603–620. https://doi.org/10.1152/jn.00528.2012 PMID: 23076101

40. Soares C, Trotter D, Longtin A, Béïque JC, Naud R. Parsing out the variability of transmission at central synapses using optical quantal analysis. Frontiers in synaptic neuroscience. 2019; 11:22. https://doi.org/10.3389/fnsyn.2019.00022 PMID: 31474847

41. Ritzau-Jost A, Delvendahl I, Rings A, Byczkowicz N, Harada H, Shigemoto R, et al. Ultrafast action potentials mediate kilohertz signaling at a central synapse. Neuron. 2014; 84(1):152–163. https://doi.org/10.1016/j.neuron.2014.08.036 PMID: 25220814

42. Saviane C, Silver RA. Fast vesicle reloading and a large pool sustain high bandwidth transmission at a central synapse. Nature. 2006; 439(7079):983–987. https://doi.org/10.1038/nature04509 PMID: 16496000

43. Bellman R. Dynamic programming. Science. 1966; 153(3731):34–37. https://doi.org/10.1126/science.153.3731.34 PMID: 17730601

44. Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.

45. Fang M, Li Y, Cohn T. Learning how to active learn: A deep reinforcement learning approach. arXiv preprint arXiv:170802383. 2017;.

46. Gontier C. Statistical approaches for synaptic characterization. University of Bern. 2022;.

47. Kutschireiter A, Surace SC, Sprekeler H, Pfister JP. Nonlinear Bayesian filtering and learning: a neuronal dynamics for perception. Scientific reports. 2017; 7(1):1–13. https://doi.org/10.1038/s41598-017-06519-y

48. Bentley JL, Saxe JB. Generating sorted lists of random numbers. ACM Transactions on Mathematical Software (TOMS). 1980; 6(3):359–364. https://doi.org/10.1145/355900.355907

49. Douc R, Cappé O. Comparison of resampling schemes for particle filtering. In: Ispa 2005. proceedings of the 4th international symposium on image and signal processing and analysis, 2005. IEEE; 2005. p. 64–69.

50. Kita K, Albergaria C, Machado AS, Carey MR, Müller M, Delvendahl I. GluA4 facilitates cerebellar expansion coding and enables associative memory formation. Elife. 2021; 10:e65152. https://doi.org/10.7554/eLife.65152 PMID: 34219651